UAV goniometric mapping and numerical integration on the Sphere

Zievi Ursin Solèr\(^1\)

\(^1\)Affiliation not available

June 07, 2024

Abstract

Possible choices of flight paths on spherical surfaces—for example on hemispheres—for goniometric mapping with UAVs are compared and explicit parametrizations are given. The quantities measured during mapping should be easily integrable numerically in order to determine physical quantities like source strength, flux, etc. Some brief considerations regarding energy budget are given also.

NOTE: “This work has been / will be submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.”
UAV goniometric mapping and numerical integration on the Sphere

Zievi Ursin Solèr; FHGR, Pulvermühlestrasse 57, Chur, Switzerland.

May 29, 2024

1 Abstract

Possible choices of flight paths on spherical surfaces - for example on hemispheres - for goniometric mapping with UAVs are compared and explicit parametrizations are given. The quantities measured during mapping should be easily integrable numerically in order to determine physical quantities like source strength, flux, etc.

Some brief considerations regarding energy budget are given also.

2 Motivation

Goniometric determination of quantities like the total radiation emitted from a point like source from a UAV does benefit from a flight path that follows a spherical surface with constant distance to the source [1, 2, 3] instead of the usual flat mapping pattern with constant distance to the ground used [15]. Due to a limited accuracy of any positioning system and thus finite acceptance radius for waypoints they must not fall below a given minimal spacing. The integration and the weights must have some tolerance against position inaccuracies. The points need to be converted to a flight path or sequence of waypoints and thus must have an ordering. Therefore the standard goniometric lattices of Latitude-Longitude type are not optimal. Preferable would be to have a constant point spacing (uniform distribution) as compact as possible with a natural ordering that allows to derive a smooth flight path.

3 Method

Quantification of the quality of investigated lattices according to the uniform distribution areas and integration weights assigned to the lattice points needs segmentation of the surface of the sphere and suitable measures.

Segmentation of sphere must result in sets of points that contain all points closer to a given lattice point than any other. The Voronoi tessellation is able to segment the spherical surface accordingly.

Uniformity of the distribution of the lattice points requires the distance between any two lattice points to be similar. Looking at a single point this would mean all other lie on a circle around it. Such an alignment cannot cover an area but the less optimal approximations given by polygons of similar angles and sides. A uniform distribution thus must result in areas that have approximately a constant extension in any direction. These shapes are in $\mathbb{R}^2$ characterized by a low perimeter compared to big area contained (optimum is circle). The equivalent on spherical surfaces are spherical caps. Compactness is the measure that allows to characterize how circle or cap like an area is. On spherical surfaces the compactness measure (also circularity, sphericity) used is the spherical isoperimetric inequality/quotient according to Paul Lévy (1919) and can be derived from a spherical cap on a sphere of radius $r$ as a function of the polygon area $A$ and the perimeter length $P$

$$\left(4\pi - \frac{A}{r^2}\right) \cdot \frac{A}{P^2}$$

With value in $[0, 1]$ where 1 is a cap and very elongated shapes approximate zero with same behavior as their area does.

The length of the perimeter can be calculated by segmenting any polygon into straight lines in $\mathbb{R}^3$ and applying the law of cosines to get the great circle arc length in radians which gives the length on the unit sphere.
The uniformity of the integration weights can be measured by comparing the area sizes for all lattice points. The area of the Voronoi tessellation patches is retrieved by dividing them into triangles and applying the method of Van Oosterom and Strackee (1983) to calculate their solid angles, as mentioned in the source documentation of [17].

4 Results

4.1 Latitude-Longitude type lattices

Using spherical coordinates where \( F \) is an arbitrary function with integral \( I \) on the sphere

\[
x = r \sin(\theta) \cos(\phi) \\
y = r \sin(\theta) \sin(\phi) \\
z = r \cos(\theta)
\]

\[
dA = r^2 \sin(\theta) \, d\theta \, d\phi = r \, d(r \cos(\theta)) \, d\phi = r \, dz \, d\phi \\
I = r^2 \int_0^\pi \int_0^{2\pi} \sin(\theta) F(r, \theta, \phi) \, d\phi \, d\theta \approx r^2 \sum_i \sum_j w_{i,j} F(r, \theta_i, \phi_j)
\]

and the approach of uniform spacing in both angles (latitude \( \theta \) and longitude \( \phi \)) a simple lattice \( \Lambda_{\text{latlon}} \) with weights \( w_{i,j} \) can be generated:

\[
\theta_i = \frac{\pi}{N_\theta} \cdot (i - \frac{1}{2}) \\
\phi_j = \frac{2\pi}{N_\phi} \cdot j \\
\Lambda_{\text{latlon}} = \{ (\theta_i, \phi_j) | i \in \{1, 2, ..., N_\theta \} \land j \in \{1, 2, ..., N_\phi \} \} \\
w_{i,j} = \frac{2\pi}{N_\phi N_\theta} \sin(\theta_i)
\]

with the total number of points \( N = N_\theta N_\phi \) (including the poles is of no use as the weight is zero). To have compact covering the correct ratio between increments in both angles has to be used, as rule of thumb \( N_\phi \approx \frac{3}{2} N_\theta \) works well.

For integration let us consider lattices with known, higher accuracy as e.g. the one given by Peirce [4], which has arbitrary polynomial precision \( 4m + 3 \) with \( m = 0, 1, ... \). The Peirce lattice \( \Lambda_{\text{peirce}} \) with weights \( w_{i,j} \) can be written as:

\[
\phi_j = \frac{2\pi}{4m+4} \cdot j, \quad A_j = \frac{2\pi}{4m+4} \\
0 = P_{2m+2}(y_i = \cos(\theta_i)), \quad B_i = \frac{1}{P_{2m+2}(y_i)} \int_{-1}^{1} y_{j} P_{2m+2}(y) \, dy \\
\Lambda_{\text{peirce}} = \{ (\theta_i, \phi_j) | i \in \{1, 2, ..., 2m + 2 \} \land j \in \{1, 2, ..., 4m + 4 \} \land m \in \{0, 1, ... \} \} \\
w_{i,j} = B_i A_j
\]

Note the \( y_j = \cos(\theta_j) \) are the \( 2m + 2 \) zeros of the Legendre polynomial \( P_{2m+2} \) of degree \( 2m + 2 \), orthogonalized on \([-1, 1]\) (these zeros can be calculated as demonstrated before in Soler [7] or similar to the \( r_k \) mentioned in sec. 10.1, furthermore they are tabulated and available from free software packages).

---

1. All calculations are done on the unit sphere, in order to work with arbitrary radii use \( r \neq 1 \) and scale the weights given by \( r^2 \).
2. Alternatively distributing the points proportional to \( dz \) instead of \( d\theta \) by using an arcsc would change the weight to \( \frac{2\pi}{N_\phi N_\theta} \) and make the scheme exact for constant functions, see eq. 2 and the Fibonacci lattice in eq. 6.
(a) Simple lattice with 32 points ($N_\theta = 4$ and $N_\phi = 8$). The integration error for a constant function is $< 2.7\%$.

(b) Peirce quadrature (surface only) with 32 points ($m = 1$) and polynomial precision of 7. Integration error for a constant function is $< 1.2 \cdot 10^{-13}\%$.

Figure 1: Latitude-Longitude type lattices. The center of the spherical caps represents the lattice point, the area the weight. The lattices are very similar for same number of points. The dashed line represents a possible flight path, as visible no natural ordering of points and thus flight path exists.

In this lattice the number of points on the surface cannot be chosen freely as it is given by $8(m + 1)^2$. Furthermore in practice it can become cumbersome or even impossible to assign the correct integration weights to the acquired data points.

Both lattices use a product Ansatz and thus lack a natural ordering, even though an ordering can be introduced artificially the flight path is not smooth and not suited for fixed-wings at least in some places. However more severe is the fact that they are made of points that are not uniformly distributed as the inter-point distance shrinks the closer the points lie to the poles. This is unpractical and ultimately impossible to follow due to finite position accuracy given by any location system.

Preferred are uniform and rotationally invariant lattices like Lebedev quadrature or point distributions retrieved from electrostatic simulation of repulsing points like charged particles. While being very uniform these schemes do have drawbacks like not being flexible in number of points or being expensive to generate.

4.2 Spiral type lattices

The approach in eq. (3) can easily be modified such that the latitude $\theta$ increases continuously along the path and by that spirals up to the pole, thus a simple spiral lattice $\Lambda_{\text{s}\text{spiral}}$ is:

$$\Lambda_{\text{s}\text{spiral}} = \{(\theta_i, \phi_i) = (\theta_j, \phi_j) | i = j \in \{1, 2, ..., N_\theta = N\}\}$$

with $N_\phi$ being the number of points per revolution. Assigning proper weights for integration is not straightforward. If $N_\phi \in \mathbb{N}$ the angle is an integral fraction and the lattice essentially is a latitude-longitude type as $(\phi_i \mod 2\pi)$ are the same on every revolution.

For a choice of $N_\phi \in \mathbb{R}\setminus\mathbb{Q}$ (irrational numbers) all $\phi_i$ are unique and the same value never appears twice. Furthermore $\theta_i$ constantly increases and thus is never the same for any two points. This allows to distribute points quite uniformly.

With $4 \leq N_\phi \leq N/2$ the spiral has at least 4 points per revolution and 1 revolution per hemisphere.
As mentioned before to have a compact covering the correct ratio between increments in both angles has to be used. The Fibonacci lattice $\Lambda_{\text{fib}}$ uses for $N_\phi$ the golden ratio and a constant spacing on the z axis connecting the poles as $dA$ is proportional to $dz$ and not $d\theta$ (see eq. (2) and Deserno [9]):

\begin{align}
\theta_i &= \arccos \left( \frac{2i - N - 1}{N} \right) \\
\phi_i &= \frac{2\pi}{N_\phi} \cdot \frac{2i - N - 1}{2}, \quad N_\phi = \frac{1 + \sqrt{5}}{2} \\
\Lambda_{\text{fib}} &= \{ (\theta_i, \phi_i) \mid i \in \{1, 2, \ldots, N\} \} \\
w_i &= \frac{4\pi}{N} \cdot 1
\end{align}

with an arbitrary number of points (odd or even). This lattice is cheap to generate. The uniformity of the lattice can be estimated from the area of the Voronoi tessellation patches using [17]. As the weights are the same for all points, positional inaccuracies - e.g. due to finite location precision - are not an issue for their assignment. Integration is done by summing at all points and finally scaling by the weight.

The points have a natural ordering (starting from the south pole) and thus a flight path can easily be constructed. However only about 2-3 points are within one revolution and thus the path becomes triangle...
shaped and penetrates deep into the volume of the sphere, it does not stay on the surface, see fig. 3. Adding intermediate waypoints increases the complexity of handling and generates a very long path (not desired due to finite battery capacity).

Figure 3: Fibonacci lattice flight path. In the equatorial plane only about 36 % of the radius is free volume. A surface only path that reduces the length to about 18 % ($N = 201$), can be generated. However it is expensive, non-smooth, non-obvious and thus cumbersome to handle. Note that the $z$ coordinate increases towards the center.

It is possible to generate a flight path that strictly stays on the surface by solving the traveling salesman problem using a genetic algorithm for the given point set as shown in fig. 3c. This also greatly reduces the length of the path but it will lead to a non-smooth pseudo random zig-zag path. However calculation of this path is expensive and not deterministic due to inherent randomness.

An Archimedean spiral on the spherical surface does have the property of equidistant spiral arms and therefore after choosing the spacing between those arms, it can be applied to separate the points along the spiral. This can be done analytically using Elliptic Integrals of first and second kind $K$ and $E$ along with a helper function $S$ as well as highly efficient fixpoint iteration schemes $[10]$: 

$$E(\phi, m) = \int_{0}^{\phi} \sqrt{1 - m \sin^2 \theta} d\theta, \quad 0 \leq \phi \leq \pi/2$$

$$E(m) = \int_{0}^{\pi/2} \sqrt{1 - m \sin^2 \theta} d\theta = E(\pi/2, m)$$

$$K(m) = \int_{0}^{\pi/2} \left(1 - m \sin^2 \theta\right)^{-1/2} d\theta$$

$$S(\theta) = \begin{cases} 
2E(-m^2) - E(\pi - \theta, -m^2) & \text{if } \pi/2 < \theta \leq \pi \\
E(\theta, -m^2) & \text{if } 0 \leq \theta \leq \pi/2 
\end{cases}$$

The lattice $\Lambda_{\text{arch}}$ can be generated by the fixpoint iterations for the parameter $m_k$ and $\theta_{j,k}$ for $k \to \infty$ (usually < 10 iterations are sufficient as they converge very fast $[10]$):
Figure 4: Archimedean lattice. The variations in compactness are small while the weights show some low values. Weights normed to analytical value of $4\pi/N$. Histogram inserts for $N=50, 500, 950$.

\[
m_{k+1} = \frac{m_k \pi N (2E(-m_k^2) - K(-m_k^2))}{N \pi E(-m_k^2) - N \pi K(-m_k^2) + m_k E(-m_k^2)^2}, \quad m_{k=0} = \sqrt{N\pi}
\]
\[
\theta_{i,k+1} = \theta_{i,k} + \frac{(2i - 1)\pi - mS(\theta_{i,k})}{m\sqrt{1 + m^2 \sin(\theta_{i,k})^2}}, \quad \theta_{i,k=0} = \arccos \left( 1 - \frac{2i - 1}{N} \right)
\]
\[
\phi_i = m_{k \to \infty} \theta_{i,k \to \infty}
\]
\[
\Lambda_{arch} = \{(\theta_{i,k \to \infty}, \phi_i) \mid i \in \{1, 2, \ldots, N\}\}
\]
\[
w_i = \frac{4\pi}{N} \cdot 1
\]

The uniformity is slightly worse than for the Fibonacci lattice in fig. 2.

To avoid the need to use several types of Elliptic Integrals and reduce the number of function evaluations a simple and efficient variant based on ref. [11] is given for convenience as lattice $\Lambda_{easyarch}$.
(a) Weights from Voronoi patch area for $N = 201$. Poles on the edges, equator in the center.

(b) Lattice points and Voronoi tessellation to determine the weights, uniformity and compactness for $N = 201$.

(c) Weights and their distribution for $N \leq 1000$.

(d) Compactness and its distribution for $N \leq 1000$.

Figure 5: Easy Archimedean lattice. Weights normed to analytical value of $4\pi/N$. Histogram inserts for $N = 50, 500, 950$.

\[ m = \pi \sqrt{\frac{N-1}{2}} \]
\[ s_i = \frac{\pi^2}{m} \cdot (i-1) \]
\[ \phi_{i,k+1} = \phi_{i,k} + \frac{s_i - E(\phi_{i,k}/m, -m^2)}{\sqrt{1/m^2} + \sin(\phi_{i,k}/m)^2}, \quad \phi_{i,k=0} = s_i \]
\[ \theta_i = \frac{\phi_{i,k \to \infty}}{m} \]
\[ \Lambda_{easymarch} = \{ (\theta_i, \phi_{i,k \to \infty}) \mid i \in \{1, 2, ..., N\} \} \]
\[ w_i = \frac{4\pi}{N} \cdot 1 \]

This approach has been tested for $4 \leq N \leq 2001$ (smaller values are not well defined or useful) and yields comparable results to the optimal one before. The iteration terminates within 10 steps on the $\varepsilon = 1.0 \cdot 10^{12}$ level.

In order to setup a flight path it is useful to define the inter-point distance $d$ and the desired radius $r$. 

7
From that estimate the number of points $N$ by using the relation

$$d = r \cdot (s_2 - s_1) = rs_2 = r\pi \sqrt{\frac{2}{N - 1}} \quad (10)$$

This allows to choose the distance between waypoints in order to fulfill acceptance radii and mission constraints.

5 Excursus: Energy budget

Compare the energy needed for a UAV flying on the Fibonacci lattice surface only flight path versus one on the Archimedean lattice. The assumption is that climbing is expensive in that it cost a lot of energy and thus the later is more energy efficient.

The flight path can be modeled piecewise as it consists of acceleration phases, linear flight, hovering at the lattice points (for data acquisition) and climbing. The energy used for this horizontal stop, hover and accelerate pattern between the waypoints should be approximately the same for every section and can thus be neglected for comparison as long as both lattices consist of the same number of points $N$. Vertical flight differs from horizontal by additionally increasing potential energy during climbing but on the other hands needs less for acceleration and stop as this can to some parts be converted from or to potential energy. Wind and other external influences are neglected. Also we consider a UAV flying slow such that the energy usage for flying a straight line is close to the one for hovering. Consider a typical medium sized UAV with following specs:

- Propellers: 21 x 7.0” x 4
- Weight: 5165 g
- Battery Capacity: 157Wh x 2
- Flight Time: 38 min

From these specs we can estimate the average power usage for hovering or slow flight:

$$P = 2 \cdot \frac{157Wh}{38/60h} \approx 496W \quad (11)$$

on the other hand is the power for climbing given by the potential energy:

$$E = mgh = 5.165kg \cdot 9.81m/s^2 \cdot 1m \approx 51J \quad (12)$$

These numbers show that the power for vertical flight at 10 m/s is comparable to hovering or slow horizontal flight, so if the UAV climbs with that rate it doubles the power usage. For moderate non-acrobatic flight the hovering clearly dominates the power usage. Alternatively hovering or slow horizontal flight for 1 s uses the same amount of energy than climbing 10 m.

To verify with another approach estimate the power needed for hovering and slow flight with a thrust-to-weight ratio of 2 ($F = 2F_g$) from refs. [13, 14, 12]:

$$P = 4 \cdot K \frac{F^{3/2}}{r} = 4 \cdot K \frac{(2F_g)^{3/2}}{r} = 4 \cdot K \frac{(2mg)^{3/2}}{r}$$

$$= 4 \cdot 0.3636m^{3/2}/kg^{1/2} \cdot \frac{(2 \cdot (5.165kg/4) \cdot 9.81m/s^2)^{3/2}}{(21inch/2) \cdot 0.0254m/inch} \approx 696W, \quad K \approx 0.3636m^{3/2}/kg^{1/2} \quad (13)$$

With this estimate hovering or slow flight becomes even more dominant of course these are average values. For a flight path on a semi-sphere of radius 10 m it is neglectable how often the UAV climbs upwards in the Fibonacci lattice (usually around 3 times in the example given in fig. 3c) as each climbing process is equivalent to hovering for an additional second (usually climbing will take more time).
Comparing the flight path length for $N = 201$ and thus the time needed for a mission using slow flight (hovering) the surface only path for the Fibonacci lattice ($\approx 50$) seems to be very optimal. The length for Archimedean spiral is $\approx 50$ and for the easy Archimedean $\approx 60$ (+ 20%). This is consistent with the fact that the same number of points distributed with a similar uniformity (weight, compactness) on the same area gives a similar inter-point distance. Connecting these points with shortest paths e.g. to nearest neighbors should always result in a similar path length. However the natural Fibonacci path as shown in fig. 3a has a length of $\approx 300$.

6 Conclusion

Several lattices have been investigated and the suitability of spiral lattices for goniometric mapping missions of UAVs underscored. The uniformity of the lattices decreases in order of their appearance in this work but at the same time the applicability increases. The Fibonacci lattice would be simple and cheap to generate with uniform weights and high compactness but is not able to provide smooth flight paths. At the other end of the scale are the Archimedean lattices that while more involved to generate are able to construct very well suited and short flight paths and also allow to a-priori estimate the inter-point distance.

A very brief look has also been done into the energy budget of these flight paths. As shown the energy used to climb up the sphere is neglectable compared to the energy used for horizontal flight or hovering and thus it does not matter whether a flight path climbs just once to the top or several times.

A very nice additional benefit of the Archimedean lattices over the Fibonacci is a deterministic generation of the flight path that delivers predictable and reproducible outcome compared to the partially random nature of the flight path generated by optimization using genetic algorithm.

7 Outlook

This purely theoretical analytical work should be supplemented by field measurements and concrete results from testing using the proposed flight paths.

Currently work is ongoing utilizing the flight paths proposed. Measurement data will include battery voltage and current data that allow for power calculation and thus eliminating the capacity from the equations.

8 Acknowledgments

I would like to express my appreciation to FHGR for funding of the project utilizing these results, METAS for coming up with the problem of flight path on a spherical surface, Ch. Bermes for some generic hints, M. Ruggia for concrete feedback and everybody that directly or indirectly contributed to this work.

9 References

References


10 Appendix

10.1 Peirce quadrature (radial part)

The approach given by Peirce [4] for the numerical integration over a spherical shell has the advantage of arbitrarily high polynomial accuracy $s = 4m + 3$ (m = 0, 1, 2, ...) but comes at the expense of having to evaluate up to $8(m + 1)^3$ points for volume integration, cheaper schemes have been reported. According to Mustard [5] the approach can be generalized to any number of dimensions $n$

$$I = \int_0^1 \int_0^\pi \int_0^{2\pi} r^2 \sin(\theta) F(r, \theta, \phi) d\phi d\theta dr = \sum_i \sum_j \sum_k D_{ijk} F(r_k, \theta_j, \phi_i)$$ (14)

For the interested reader follows an explicit derivation of the radial lattice $(r_k^2, C_k)$ with $k = 1, 2, ..., m + 1$ as this is a Gauss Quadrature rule. The calculation can be done numerically, analytically and even for arbitrarily high polynomial accuracy

Using the monomials $\varphi_j(x) = x^{j-1}$ and the weight function $\omega(x) = \sqrt{x}$ construct the momentum matrix ("Hankel" matrix in the moments)

$$M = \{m_{i,j}\} = [\varphi_i, \varphi_j]_{i,j=1}^{N+1} = \left[\int_{R^2} \omega(x)\varphi_i(x)\varphi_j(x)dx\right]_{i,j=1}^{N+1}$$ (15)

with $0 \leq R \leq 1$. From the Cholesky decomposition $M = U^T U$ and $U = \{u_{i,j}\}$ construct the recurrence relation matrix

$$J = \begin{bmatrix}
\alpha_1 & \beta_1 & 0 \\
\beta_1 & \alpha_2 & \beta_2 \\
0 & \ldots & \ldots \\
0 & \ldots & \beta_{N-1} \\
\beta_{N-1} & \alpha_N & 0
\end{bmatrix}, \quad \alpha_j = \frac{u_{j,j+1}}{u_{j,j}} - \frac{u_{j-1,j}}{u_{j-1,j-1}}, \quad \beta_j = \frac{u_{j+1,j+1}}{u_{j,j}}$$ (16)

with $u_{0,0} = 1$ and $u_{0,1} = 0$. The eigenvalues of $J$ are the $r_k^2$ and the weights are $2 \cdot C_k = q_{1,j}^2 \cdot m_{0,0}$ given by $q_{1,j}$ the first component of the $j$-th orthonormal eigenvector and the first moment $m_{0,0}$. Also note that the $r_k^2$ are the zeros of the orthonormal polynomials $Q_{m+1}(r^2) = \prod_k (r^2 - r_k^2)$ for the quadrature mentioned by Peirce [4].

Note the combined weights $D_{ij} = A_i B_j = w_{j,i}$ (indices $i$ and $j$ swapped) allow for the integration over the surface of the unit sphere (sec. 4.1), while $D_{ijk} = A_i B_j C_k$ for volume integration over a spherical shell, these weights add up to the surface area and volume accordingly (Mustard [5]).

To prove the performance of this algorithm, the weights, zeros and polynomials for $m = 0$ and $m = 1$ given by Peirce [4] have been generalized to an arbitrary $R$ and are listed in the following. The case for $m = 0$ (and arbitrary $n$) as given by Mustard [5] is provided for verification.

$$m = N - 1 = 0$$

$$r_1^2 = \frac{3 (R^5 - 1)}{5 (R^3 - 1)}$$ (17)

$$C_1 = -\frac{(R^3 - 1)}{3}$$ (18)

$$Q_1(x) = x - \frac{3 (R^5 - 1)}{5 (R^3 - 1)}$$ (19)

$$m = N - 1 = 1$$

11
\[ a = 512R^{14} + 5120R^{13} + 28160R^{12} + 103840R^{11} + 278080R^{10} + 559168R^9 + 850875R^8 + 978990R^7 + 850875R^6 + 559168R^5 + 278080R^4 + 103840R^3 + 28160R^2 + 512R + 512 \] (20)

\[ b = 280R^8 + 1120R^7 + 2800R^6 + 4375R^5 + 4900R^4 + 4375R^3 + 2800R^2 + 1120R + 280 \]

\[ c = 4R^6 + 16R^5 + 40R^4 + 55R^3 + 40R^2 + 16R + 4 \]

\[ r_1^2 = \frac{b + \sqrt{35}(R - 1)\sqrt{a}}{126c}, \quad r_2^2 = \frac{b - \sqrt{35}(R - 1)\sqrt{a}}{126c} \] (21)

\[ C_1 = \frac{-2268(R^3 - 1)\left(25(R^3 - 1)(R^7 - 1) - 21(R^5 - 1)^2\right)c^2}{6804\left(25(R^3 - 1)(R^7 - 1) - 21(R^5 - 1)^2\right)c^2 + (5(R^3 - 1)(b + \sqrt{35}(R - 1)\sqrt{a}) - 378(R^5 - 1)c)^2} \]

\[ C_2 = \frac{-2268(R^3 - 1)\left(25(R^3 - 1)(R^7 - 1) - 21(R^5 - 1)^2\right)c^2}{6804\left(25(R^3 - 1)(R^7 - 1) - 21(R^5 - 1)^2\right)c^2 + (5(R^3 - 1)(b - \sqrt{35}(R - 1)\sqrt{a}) - 378(R^5 - 1)c)^2} \] (22)

\[ Q_1(x) = x^2 - \frac{b}{63c}x + \frac{60R^{10} + 240R^9 + 600R^8 + 1200R^7 + 2100R^6 + 2625R^5 + 2100R^4 + 1200R^3 + 600R^2 + 240R + 60}{63c} \] (23)

Note that this scheme was reported to have failed to produce the recursion coefficients numerically from the moments when \( N \) was about 20 for some cases (Golub [6]). Also the analytic solution for arbitrary \( R \) is big and cumbersome to handle for \( m > 1 \), thus it is recommended to use explicit \( R \) when calculating the analytical solution.