Software for fast and accurate evaluation of geomagnetic quantities at scattered points in space

Pencho Petrushev, Kamen Ivanov

University of South Carolina[1]

Introduction

This readme file contains a description of the MATLAB code

<div align="center">

`emmsynth_fast.m`

</div>

for fast and accurate evaluation of all components of the geomagnetic field represented in high degree (> 720) solid spherical or ellipsoidal harmonics at many scattered points in the space above the surface of the Earth.

The evaluated magnetic field components are listed in Subsection 2.2. Their values are derived from the official NOAA Enhanced Magnetic Models EMM2015 or EMM2017. In what follows we utilize the terminology and notation from the description of the World Magnetic Model WMM2015, given in `WMM2015_Report.pdf` at `https://www.ngdc.noaa.gov/geomag/WMM/`.

The package `emmsynth_fast_m.zip` contains all files necessary for the execution of the code. Before using the main code `emmsynth_fast.m` one has to initialize it by running `emmsynth_init.m`. The code `emmsynth_init.m` uses the EMM2015 or EMM2017 coefficients given in `EMM2015.COF` and `EMM2015SV.COF` or `EMM2017.COF` and `EMM2017SV.COF` and produces six binary data files `emm2015_*.bin` or `emm2017_*.bin` (of 1.5 GB each, not included in the package) with the weighted values of the magnetic gradient and their secular variations at regular points located on a number of confocal ellipsoids. These data files are input files for `emmsynth_fast.m` together with the applied magnetic model, the decimal year for model evaluation and the user defined file `scattered_points.dat`. The package contains two test files `scattered_points.dat` with 1,000 and 1,000,000 point coordinates, respectively. The expected test results of the 1,000 point file are collected in **scattered_points_values1000_EMM2015_2018.zip** for magnetic model EMM2015 and year 2018 or `scattered_points_values1000_EMM2017_2019.zip` for magnetic model EMM2017 and year 2019.

The purpose of the third code `emmsynth_standard.m` is to monitor the accuracy of the performance and to serve as a benchmark for the speed of the code `emmsynth_fast.m`. The code `emmsynth_standard.m` uses the file `scattered_points_values.dat` as an input file and evaluates the magnetic field components applying the standard methods for spherical harmonic computation based of the model coefficients included in `EMM2015.COF` and `EMM2015SV.COF` or `EMM2017.COF` and `EMM2017SV.COF`.

The current file describes (§4) the key components of the algorithm.

1 Files in the package

The package `emmsynth_fast_m.zip` contains the following files:

- `emmsynth_fast.m` containing the MATLAB code of the main program;

- `emmsynth_standard.m` containing the MATLAB code of the testing program;

- `emmsynth_init.m` containing the MATLAB code for initialization of `emmsynth_fast.m` with the binary data files `emm2015_*.bin` or `emm2017_*.bin`.

- `EMM2015.COF` and `EMM2015SV.COF` are the free format ASCII files containing all EMM2015 coefficients. `EMM2017.COF` and `EMM2017SV.COF` are the free format ASCII files containing the coefficients from the predictive part of EMM2017. The coefficient files have been downloaded from

  `https://www.ngdc.noaa.gov/geomag/EMM/`

  These two couples of files are used by `emmsynth_init.m` and `emmsynth_standard.m`.

- Archives `scattered_points1000.zip`, `scattered_points1000000.zip` containing the file `scattered_points.dat` with 1,000 and 1,000,000 point coordinates, respectively, as test input files;

- Each of the archives `scattered_points_values1000_EMM2015_2018.zip` and `scattered_points_values1000_EMM2017_2019.zip` contains the two 1,000 point test output files `scattered_points_values.dat` and `scattered_points_values_standard.dat` computed for the respective model (EMM2015 or EMM2017) and year (2018 or 2019) for all magnetic field components;

- `readme_emmsynth_fast.pdf` is the current file.

2 Input and output files

2.1 Free format ASCII input file

The ASCII input file `scattered_points.dat` created by the user contains the geodetic geographic coordinates of the scattered points in space, where the magnetic field is to be evaluated. The file has one record for every point. Each record contains the coordinates of one point: the geodetic geographic latitude and the longitude in decimal degrees followed by the height above the reference ellipsoid in meters.

```
    (decimal degrees)   (decimal degrees)      (meters)
    geodetic latitude        longitude     geodetic distance
```

These data are read using free FORMAT. Geodetic coordinates should refer to the WGS84. The scattered points have to be located in the range from -415 up to $1,000,000$ meters above the Earth's reference ellipsoid.

2.2 Free format ASCII output files

The ASCII output file `scattered_points_values.dat` contains the magnetic field components' values at the scattered points from the input. The file has one record with 13 entries for every point. The first 3 entries are the point coordinate:

1. Geodetic geographic latitude (in decimal degrees)

2. Geodetic geographic longitude (in decimal degrees)

3. The height above the reference ellipsoid (in meters)

as read from `scattered_points.dat`.

The last 10 entries are the values at the above point of the following components of the magnetic field computed by `emmsynth_fast.m` (numbered according to their position in the record):

4. North component in geocentric coordinates X' (in nT 'nanoTesla')

5. East component in geocentric coordinates Y' (in nT)

6. Down component in geocentric coordinates Z' (in nT)

7. North component in geodetic coordinates X (in nT)

8. East component in geodetic coordinates Y (in nT)

9. Down component in geodetic coordinates Z (in nT)

10. Horizontal intensity H (in nT)

11. Total intensity F (in nT)

12. Inclination I (in decimal degrees)

13. Declination D (in decimal degrees)

The file `scattered_points_values.dat` also serves as input for `emmsynth_standard.m`.

The ASCII output file `scattered_points_values_standard.dat` has the same structure as `scattered_points_values.dat`, where the last 10 entries of every record are the values computed by `emmsynth_standard.m` (instead of `emmsynth_fast.m`).

2.3 The EMM2015 coefficients input files

`EMM2015.COF` and `EMM2015SV.COF` are the files containing the standard EMM2015 coefficients. `EMM2017.COF` and `EMM2017SV.COF` are the files containing the coefficients from the predictive part of EMM2017. They have been downloaded from

`https://www.ngdc.noaa.gov/geomag/EMM/`.

These files are required as input by `emmsynth_init.m` and `emmsynth_standard.m`.

2.4 Binary data files

The six data files `emm2015_*.bin` or `emm2017_*.bin` are not included in the package. They are produced by `emmsynth_init.m` in order to initialize `emmsynth_fast.m`. Every file contains several records – one record for every confocal ellipsoid used by the code (see §4.5). Every unformatted record contains double precision numbers with little-endian byte ordering. These numbers are the weighted values of the respective magnetic field components at regular points on the corresponding ellipsoid.

The binary data files for EMM2015 and their sizes are

- `emm2015_Xp.bin` (1.5 GB);

- `emm2015_Yp.bin` (1.5 GB);

- `emm2015_Zp.bin` (1.5 GB);

- `emm2015_XpSV.bin` (1.5 GB);

- `emm2015_YpSV.bin` (1.5 GB);

- `emm2015_ZpSV.bin` (1.5 GB).

The sizes of the 6 data files for EMM2017 are the same, while their names are appropriately modified. Besides the applicable magnetic model the file names also contain indication of the magnetic field components X', Y', Z' and the respective secular variation quantities.

3   How to run the codes

1. Run `emmsynth_init.m` with the choice of the model (enter 1 for EMM2015 or 2 for EMM2017) to generate the six data files `emm2015_*.bin` or `emm2017_*.bin`, respectively.

2. Create a free format ASCII input file `scattered_points.dat` with the coordinates of the scattered points.

3. Run `emmsynth_fast.m` on `scattered_points.dat` with the choice of the model (enter 1 for EMM2015 or 2 for EMM2017) and with the decimal year in the range from 2015 to 2020 for EMM2015 or in the range from 2017 to 2022 for EMM2017 as input parameters in order to obtain the ASCII output file `scattered_points_values.dat`. One may repeat many times the previous and the current steps for different scattered point files, models and years without performing Step 1.

4. If you want to check the accuracy of `emmsynth_fast.m`, then run `emmsynth_standard.m` as on the `emmsynth_fast.m` output `scattered_points_values.dat` with the same choices of model and year.

Remarks:

- All data files are located in the same files folder as the codes.

- Every data file `emm2015_*.bin` is 1.5 GB and all 6 files are simultaneously loaded by `emmsynth_fast.m` when EMM2015 is chosen. The same for EMM2017.

- `emmsynth_init.m` works 11 minutes on my computer for a chosen model.

- When `emmsynth_standard.m` is run after `emmsynth_fast.m` one should use the input file `scattered_points.dat` with no more than 60,000 points due to the slow speed of the "standard" code. (On the standard laptop we use (see §8) `emmsynth_standard.m` works more than 10 minutes to handle 60,000 points.)

4 Key components of the algorithm

4.1 Magnetic field component evaluation: Overview

Following `WMM2015_Report.pdf` we sketch the procedure for computing the magnetic field elements at a given location and time (λ, φ, h, t), where λ and φ are the geodetic longitude and latitude, h is the height above the WGS84 ellipsoid, and t is the time given in decimal years.

1. The user provided geodetic coordinates (λ, φ, h) are transformed into spherical geocentric coordinates (λ, φ', r).

2. The Gauss coefficients $g_n^m(t)$, $h_n^m(t)$ are determined for the desired time t from the model coefficients $g_n^m(t_0)$, $h_n^m(t_0)$ (read from `EMM2015.COF` or `EMM2017.COF`) and $\dot{g}_n^m(t_0)$ and $\dot{h}_n^m(t_0)$ (read from `EMM2015SV.COF` or `EMM2017SV.COF`) by

$$g_n^m(t) = g_n^m(t_0) + (t - t_0)\dot{g}_n^m(t_0), \quad h_n^m(t) = h_n^m(t_0) + (t - t_0)\dot{h}_n^m(t_0),$$

   where the time t is given in decimal years and $t_0 = 2015.0$ (or $t_0 = 2017.0$) is the reference epoch of the model.

3. The field vector components $X' = -\dfrac{1}{r}\dfrac{\partial V}{\partial \varphi'}$, $Y' = -\dfrac{1}{r\cos\varphi'}\dfrac{\partial V}{\partial \lambda}$, $Z' = \dfrac{\partial V}{\partial r}$ are computed from the scalar potential V. This potential can be expanded in terms of spherical harmonics as

$$V(\lambda, \varphi', r, t) = a\sum_{n=1}^{N}\left(\frac{a}{r}\right)^{n+1}\sum_{m=0}^{n}(g_n^m(t)\cos m\lambda + h_n^m(t)\sin m\lambda)\breve{P}_n^m(\sin\varphi'),$$

   where N is the degree of the expansion of the model ($N = 740$ for EMM2015, $N = 790$ for EMM2017), a is the *geomagnetic reference* radius ($a = 6,371,200$ m), $g_n^m(t)$ and $h_n^m(t)$ are the time-dependent Gauss coefficients of degree n and order m from Step 2, $\breve{P}_n^m$ are the Schmidt semi-normalized associated Legendre functions.

4. The geocentric magnetic field vector components X', Y', Z' are rotated into the ellipsoidal reference frame to the magnetic field vector components $X, Y = Y', Z$.

5. The magnetic elements H, F, I, D are computed from the orthogonal components X, Y, Z:

$$H = \sqrt{X^2 + Y^2}, \quad F = \sqrt{X^2 + Y^2 + Z^2},$$

$$D = \arctan Y/X, \quad I = \arctan Z/H.$$

The third step in the above algorithm is the most important and time consuming one. For this step we utilize the algorithm for fast evaluation of quantities represented in high degree solid spherical harmonics from [2], where this algorithms is applied to evaluation of gravimetric quantities in the gravitational model EGM2008. The computations in the remaining four steps are straightforward and essential improvement in time is not possible.

4.2 Comparison between the representations of gravimetric and magnetic potentials

Using the same geocentric coordinates with $\theta' = \pi/2 - \varphi'$ denoting the co-latitude we represent the disturbing gravitational potential by

$$T(\lambda, \theta', r) = \frac{GM}{a} \sum_{n=2}^{N} \left(\frac{a}{r}\right)^{n+1} \sum_{m=0}^{n} (c_{nm} \cos m\lambda + s_{nm} \sin m\lambda) \bar{P}_{nm}(\cos \theta')$$

where GM is the Earth's gravitational constant, N is the degree of the expansion of the model ($N = 2190$ for EGM2008), a is a scaling factor associated with the coefficients that is usually chosen to be numerically equal to the semi-major axis of the adopted reference ellipsoid ($a = a_e = 6,378,137$ m in WGS84), c_{nm} and s_{nm} are the fully-normalized spherical harmonic coefficients of degree n and order m describing the Earth's gravitational field, $\bar{P}_{nm}$ are the fully-normalized associated Legendre functions (ALF).

In comparing the above representations of the disturbing gravimetric potential T and the magnetic potential V we would like to make the following observations:

1. *Use of latitude and co-latitude.* This creates small problems as the argument of the ALF's is the same ($\sin \varphi' = \cos \theta'$). Differences occur in:

   - the North component of the vector field, as: $\dfrac{\partial}{\partial \varphi'} = -\dfrac{\partial}{\partial \theta'}$,

   - vector field rotation from sphere to ellipse in step 4: $\varphi' - \varphi = -(\theta' - \theta)$.

2. *The ALF normalization.* $\check{P}_n^m$ are the Schmidt semi-normalized ALF, while $\bar{P}_{nm}$ are the fully-normalized ALF. Note that ALF have the same definition in both models, see e.g. Heiskanen and Moritz, 1967. The relation is:
   $$\sqrt{2n+1}\check{P}_n^m(v) = \bar{P}_{nm}(v), \quad 0 \le m \le n, \quad n = 0, 1, \ldots.$$

3. *The scaling factor associated with the coefficients.* In the magnetic model $a = 6,371,200$ m, while $a = a_e = 6,378,137$ m in WGS84. The difference is $6,937$ m.

4. *Type of the local coordinate system.* The elements of the main magnetic field are computed with respect to a local geodetic system – the "north"

8

direction is tangential to the Earth related ellipsoid passing through the point and the "down" direction is normal to the ellipsoid (the corresponding values of the field are the Northerly intensity X and the Vertical intensity Z). Some quantities of the gravitational field are computed with respect to local spherical system – the "north" direction is tangential to the sphere passing through the point and the "down" direction is normal to the sphere (the corresponding values of the field are the north-south deflection of the vertical $\dfrac{1}{r\gamma}\dfrac{\partial T}{\partial \theta'}$, and $-\dfrac{\partial T}{\partial r}$ as gravity disturbance).

5. *Directions of the local coordinate system.* $\vec{Oz}$ points down in the magnetic case and $\vec{Oz}$ points up in the gravitational case.

6. $n \geq 1$ *or* $n \geq 2$. This difference is not important from computational view point. Here V represent the whole magnetic potential, while T is the *disturbing* gravity potential.

   In the gravity case the potential is represented by a normal part and a disturbing part T. Only the disturbing potential is modeled and computed in spherical harmonics. The contribution of the gravity disturbance to the entire gravity is approximately 0.1%!

   In the magnetic case we do not have a decomposition into a normal part and a disturbing part. Even if one considers all harmonics of degree ≤ 15 as the normal part of EMM2015, then the contribution of the remaining terms (the tail of the series) to the magnetic field components at the reference ellipsoid will amount to 8% for X, 21% for Y and 10% for Z and F. For EMM2017 the corresponding contributions amount to 12% for X, 29% for Y and 14% for Z and F.

7. $g_n^m(t)$ and $h_n^m(t)$ are in nT (nano-Tesla), while c_{nm} and s_{nm} are unit-less.

Finally we observe that from computational point of view:

- The north component in geocentric coordinates X' corresponds to the north-south deflection of the vertical ξ (in the gravitational model);

- The east component in geocentric coordinates Y' corresponds to the east-west deflection of the vertical η;

- The down component in geocentric coordinates Z' corresponds to the gravity disturbance δg.

4.3 Representation of the magnetic field components

In this part we show that Z' has a representation of the form

$$\kappa(r)\mathcal{F}(\lambda, \theta', r),$$

while X' and Y' have representations of the form

$$\kappa(r)(\mathcal{F}_1(\lambda, \theta', r) \cos \lambda + \mathcal{F}_2(\lambda, \theta', r) \sin \lambda),$$

where $\mathcal{F}, \mathcal{F}_1, \mathcal{F}_2$ are harmonic functions and κ's are simple smooth functions.

In the representations in this subsection we do not indicate the dependance on t, which comes with the coefficients c_{nm}, s_{nm} (given below).

4.3.1 From Schmidt semi-normalized to fully-normalized ALF

The magnetic scalar potential V has the following representations in the EMM2015 and EMM2017 in terms of the fully-normalized ALF (used in our algorithm):

$$V(\lambda, \varphi', r, t) = a \sum_{n=1}^{N} \left(\frac{a}{r}\right)^{n+1} \sum_{m=0}^{n} (g_n^m(t) \cos m\lambda + h_n^m(t) \sin m\lambda) \check{P}_n^m(\sin \varphi')$$

$$= a \sum_{n=1}^{N} \left(\frac{a}{r}\right)^{n+1} \sum_{m=0}^{n} (c_{nm} \cos m\lambda + s_{nm} \sin m\lambda) \bar{P}_{nm}(\cos \theta')$$

with

$$c_{nm} = \frac{g_n^m(t)}{\sqrt{2n+1}}, \quad s_{nm} = \frac{h_n^m(t)}{\sqrt{2n+1}}.$$

4.3.2 Down component

The representation of Z' in the fully-normalized ALF takes the form:

$$Z'(\lambda, \varphi', r) = \frac{\partial}{\partial r} V(\lambda, \varphi', r)$$

$$= -\sum_{n=1}^{N} (n+1) \left(\frac{a}{r}\right)^{n+2} \sum_{m=0}^{n} (c_{nm} \cos m\lambda + s_{nm} \sin m\lambda) \bar{P}_{nm}(\cos \theta').$$

Form for *spherical polynomial* computations:

$$Z'(\lambda, \varphi', r) = \kappa(r)\mathcal{F}(\lambda, \theta', r),$$

where

$$\mathcal{F}(\lambda, \theta', r) = \sum_{n=1}^{N} \left(\frac{a}{r}\right)^{n+1} \sum_{m=0}^{n} (a_{nm} \cos m\lambda + b_{nm} \sin m\lambda) \bar{P}_{nm}(\cos \theta'),$$

$$\kappa(r) = \frac{a}{r}, \quad a_{nm} = -(n+1)c_{nm}, \quad b_{nm} = -(n+1)s_{nm}.$$

Form for *ellipsoidal harmonic* computations:

$$Z'(\lambda, \varphi', r) = -\frac{a}{r} \sum_{n=1}^{N} \left(\frac{a_e}{r}\right)^{n+1}$$

$$\times \sum_{m=0}^{n} (n+1) \left(\frac{a}{a_e}\right)^{n+1} (c_{nm} \cos m\lambda + s_{nm} \sin m\lambda) \bar{P}_{nm}(\cos \theta')$$

$$= \kappa(r)\mathcal{F}(\lambda, \theta', r),$$

where

$$\mathcal{F}(\lambda, \theta', r) = \kappa \sum_{n=1}^{N} \left(\frac{a_e}{r}\right)^{n+1} \sum_{m=0}^{n} (a_{nm} \cos m\lambda + b_{nm} \sin m\lambda) \bar{P}_{nm}(\cos \theta'),$$

$$\kappa(r) = \frac{a}{r}, \quad a_{nm} = -(n+1) \left(\frac{a}{a_e}\right)^{n+1} c_{nm}, \quad b_{nm} = -(n+1) \left(\frac{a}{a_e}\right)^{n+1} s_{nm}.$$

4.3.3 North component

The representation of X' in the fully-normalized ALF is of the form:

$$X'(\lambda, \varphi', r) = -\frac{1}{r} \frac{\partial}{\partial \varphi'} V(\lambda, \varphi', r)$$

$$= \sum_{n=1}^{N} \left(\frac{a}{r}\right)^{n+2} \sum_{m=0}^{n} (c_{nm} \cos m\lambda + s_{nm} \sin m\lambda) \frac{d}{d\theta'} \bar{P}_{nm}(\cos \theta')$$

$$= \sum_{n=1}^{N} \left(\frac{a}{r}\right)^{n+2} \sum_{m=0}^{n} (c_{nm} \cos m\lambda + s_{nm} \sin m\lambda)(\alpha_{n,m} \bar{P}_{n,m+1} + \beta_{n,m} \bar{P}_{n,m-1})$$

with

$$\alpha_{n,m} = -\frac{1}{2}\sqrt{(n+m+1)(n-m)}, \quad 0 \le m \le n,$$

$$\beta_{n,0} = 0, \quad \beta_{n,m} = \frac{1}{2}\sqrt{(1+\delta_{1,m})(n-m+1)(n+m)}, \quad 1 \le m \le n.$$

Note that $\alpha_{n,n} = 0$ and $\beta_{n,0} = 0$.

Form for *spherical polynomial* computations:

$$X'(\lambda, \varphi', r) = \kappa(r)(\mathcal{F}_1(\lambda, \theta', r) \cos \lambda + \mathcal{F}_2(\lambda, \theta', r) \sin \lambda),$$

where $\kappa(r) = a/r$ and for $j = 1, 2$

$$\mathcal{F}_j(\lambda, \theta', r) = \sum_{n=1}^{N} \left(\frac{a}{r}\right)^{n+1} \sum_{m=0}^{n} (a_{n,m}^j \cos m\lambda + b_{n,m}^j \sin m\lambda) \bar{P}_{n,m}(\cos \theta')$$

with coefficients a_{nm}^j, b_{nm}^j given in Table 1.

11

$$
a_{n,m}^1 = \begin{cases} \frac{1}{\sqrt{2}}\sqrt{n(n+1)}c_{n,1}, & \text{for } m = 0; \\ \frac{1}{2}\sqrt{(n-1)(n+2)}c_{n,2} - \frac{1}{\sqrt{2}}\sqrt{n(n+1)}c_{n,0}, & \text{for } m = 1; \\ \frac{1}{2}\sqrt{(n-m)(n+m+1)}c_{n,m+1} & \\ \quad - \frac{1}{2}\sqrt{(n-m+1)(n+m)}c_{n,m-1}, & \text{for } 2 \le m \le n; \end{cases}
$$

$$
b_{n,m}^1 = \begin{cases} 0, & \text{for } m = 0; \\ \frac{1}{2}\sqrt{(n-1)(n+2)}s_{n,2}, & \text{for } m = 1; \\ \frac{1}{2}\sqrt{(n-m)(n+m+1)}s_{n,m+1} & \\ \quad - \frac{1}{2}\sqrt{(n-m+1)(n+m)}s_{n,m-1}, & \text{for } 2 \le m \le n; \end{cases}
$$

$$
a_{n,m}^2 = \begin{cases} \frac{1}{\sqrt{2}}\sqrt{n(n+1)}s_{n,1}, & \text{for } m = 0; \\ \frac{1}{2}\sqrt{(n-1)(n+2)}s_{n,2}, & \text{for } m = 1; \\ \frac{1}{2}\sqrt{(n-m)(n+m+1)}s_{n,m+1} & \\ \quad + \frac{1}{2}\sqrt{(n-m+1)(n+m)}s_{n,m-1}, & \text{for } 2 \le m \le n; \end{cases}
$$

$$
b_{n,m}^2 = \begin{cases} 0, & \text{for } m = 0; \\ -\frac{1}{2}\sqrt{(n-1)(n+2)}c_{n,2} - \frac{1}{\sqrt{2}}\sqrt{n(n+1)}c_{n,0}, & \text{for } m = 1; \\ -\frac{1}{2}\sqrt{(n-m)(n+m+1)}c_{n,m+1} & \\ \quad - \frac{1}{2}\sqrt{(n-m+1)(n+m)}c_{n,m-1}, & \text{for } 2 \le m \le n. \end{cases}
$$

Table 1: Coefficients in the representations of X'

Form for *ellipsoidal harmonic* computations:

$$
X'(\lambda, \varphi', r)
$$

$$
= \kappa(r) \sum_{n=1}^{N} \left(\frac{a_e}{r}\right)^{n+1} \sum_{m=0}^{n} \left(\frac{a}{a_e}\right)^{n+1} (c_{n,m} \cos m\lambda + s_{n,m} \sin m\lambda)
$$

$$
(\alpha_{n,m} \bar{P}_{n,m+1}(\cos\theta') + \beta_{n,m} \bar{P}_{n,m-1}(\cos\theta'))
$$

$$
= \kappa(r) \sum_{n=1}^{N} \left(\frac{a_e}{r}\right)^{n+1} \left[\sum_{m=1}^{n} \left(\frac{a}{a_e}\right)^{n+1} (c_{n,m-1} \cos(m-1)\lambda + s_{n,m-1} \sin(m-1)\lambda)\alpha_{n,m-1} \bar{P}_{n,m} \right.
$$

$$
\left. + \sum_{m=0}^{n-1} \left(\frac{a}{a_e}\right)^{n+1}, (c_{n,m+1} \cos(m+1)\lambda + s_{n,m+1} \sin(m+1)\lambda)\beta_{n,m+1} \bar{P}_{n,m} \right]
$$

$$
= \kappa(r)(\mathcal{F}_1(\lambda, \theta', r) \cos\lambda + \mathcal{F}_2(\lambda, \theta', r) \sin\lambda),
$$

where $\kappa(r) = a/r$ and for $j = 1, 2$

$$
\mathcal{F}_j(\lambda, \theta', r) = \sum_{n=1}^{N} \left(\frac{a_e}{r}\right)^{n+1} \sum_{m=0}^{n} (\bar{a}^j_{n,m} \cos m\lambda + \bar{b}^j_{n,m} \sin m\lambda) \bar{P}_{n,m}(\cos\theta')
$$

with coefficients

$$
\bar{a}^j_{nm} = \left(\frac{a}{a_e}\right)^{n+1} a^j_{nm}, \quad \bar{b}^j_{nm} = \left(\frac{a}{a_e}\right)^{n+1} b^j_{nm}, \quad j = 1, 2, \tag{1}
$$

and coefficients a^j_{nm}, b^j_{nm} given in Table 1. Note that c_{nm} and s_{nm} from Table 1 are multiplied by $(a/a_e)^{n+1}$ as in the cases of Z'.

4.3.4   East component

The representation of Y' in the fully-normalized ALF takes the form:

$$
Y'(\lambda, \varphi', r) = -\frac{1}{r \cos\varphi'} \frac{\partial}{\partial\lambda} V(\lambda, \varphi', r)
$$

$$
= \sum_{n=1}^{N} \left(\frac{a}{r}\right)^{n+2} \sum_{m=1}^{n} (c_{nm} \sin m\lambda - s_{nm} \cos m\lambda) \frac{m\bar{P}_{nm}(\cos\theta')}{\sin\theta'}
$$

$$
= \sum_{n=1}^{N} \left(\frac{a}{r}\right)^{n+2} \sum_{m=1}^{n} (c_{nm} \sin m\lambda - s_{nm} \cos m\lambda)(\alpha_{n,m} \bar{P}_{n-1,m+1} + \beta_{n,m} \bar{P}_{n-1,m-1})
$$

with

$$
\alpha_{n,m} = \frac{1}{2} \sqrt{\frac{2n+1}{2n-1}} \sqrt{(n-m-1)(n-m)}, \quad 1 \le m \le n,
$$

$$
\beta_{n,m} = \frac{1}{2} \sqrt{\frac{2n+1}{2n-1}} \sqrt{(1+\delta_{1,m})(n+m-1)(n+m)}, \quad 1 \le m \le n.
$$

$$a_{n,m}^1 = \begin{cases} -\sqrt{2(n+1)(n+2)}s_{n+1,1}, & \text{for } m = 0; \\ -\sqrt{(n+2)(n+3)}s_{n+1,2}, & \text{for } m = 1; \\ -\sqrt{(n+m+1)(n+m+2)}s_{n+1,m+1} \\ \quad -\sqrt{(n-m+1)(n-m+2)}s_{n+1,m-1}, & \text{for } 2 \le m \le n; \end{cases}$$

$$b_{n,m}^1 = \begin{cases} 0, & \text{for } m = 0; \\ \sqrt{(n+2)(n+3)}c_{n+1,2}, & \text{for } m = 1; \\ \sqrt{(n+m+1)(n+m+2)}c_{n+1,m+1} \\ \quad +\sqrt{(n-m+1)(n-m+2)}c_{n+1,m-1}, & \text{for } 2 \le m \le n; \end{cases}$$

$$a_{n,m}^2 = \begin{cases} \sqrt{2(n+1)(n+2)}c_{n+1,1}, & \text{for } m = 0; \\ \sqrt{(n+2)(n+3)}c_{n+1,2}, & \text{for } m = 1; \\ \sqrt{(n+m+1)(n+m+2)}c_{n+1,m+1} \\ \quad -\sqrt{(n-m+1)(n-m+2)}c_{n+1,m-1}, & \text{for } 2 \le m \le n; \end{cases}$$

$$b_{n,m}^2 = \begin{cases} 0, & \text{for } m = 0; \\ \sqrt{(n+2)(n+3)}s_{n+1,2}, & \text{for } m = 1; \\ \sqrt{(n+m+1)(n+m+2)}s_{n+1,m+1} \\ \quad -\sqrt{(n-m+1)(n-m+2)}s_{n+1,m-1}, & \text{for } 2 \le m \le n. \end{cases}$$

Table 2: Coefficients in the representations of Y'

Form for *spherical polynomial* computations:

$$Y'(\lambda, \varphi', r)$$
$$= \kappa(r) \sum_{n=0}^{N-1} \left(\frac{a}{r}\right)^{n+1} \sum_{m=1}^{n+1} (c_{n+1,m} \sin m\lambda - s_{n+1,m} \cos m\lambda)(\alpha_{n+1,m}\bar{P}_{n,m+1} + \beta_{n+1,m}\bar{P}_{n,m-1})$$
$$= \kappa(r)(\mathcal{F}_1(\lambda, \theta', r) \cos \lambda + \mathcal{F}_2(\lambda, \theta', r) \sin \lambda),$$

where $\kappa(r) = (a/r)^2$ and for $j = 1, 2$

$$\mathcal{F}_j(\lambda, \theta', r) = \sum_{n=0}^{N-1} \left(\frac{a}{r}\right)^{n+1} \sum_{m=0}^{n} (a_{n,m}^j \cos m\lambda + b_{n,m}^j \sin m\lambda)\bar{P}_{n,m}(\cos \theta')$$

with coefficients a_{nm}^j, b_{nm}^j given in Table 2. This representation holds in view of $\alpha_{n,n} = \alpha_{n,n-1} = 0$.

Form for *ellipsoidal harmonic* computations:

$$Y'(\lambda, \varphi', r)$$
$$= \left(\frac{a}{r}\right)^2 \sum_{n=1}^{N} \left(\frac{a_e}{r}\right)^n \sum_{m=1}^{n} \left(\frac{a}{a_e}\right)^n (c_{nm} \sin m\lambda - s_{nm} \cos m\lambda)\frac{m\bar{P}_{nm}(\cos \theta')}{\sin \theta'}$$

$$= \left(\frac{a}{r}\right)^2 \sum_{n=0}^{N-1} \left(\frac{a_e}{r}\right)^{n+1} \sum_{m=1}^{n+1} \left(\frac{a}{a_e}\right)^{n+1} (c_{n+1,m} \sin m\lambda - s_{n+1,m} \cos m\lambda)$$

$$(\alpha_{n+1,m} \bar{P}_{n,m+1}(\cos\theta') + \beta_{n+1,m} \bar{P}_{n,m-1}(\cos\theta'))$$

$$= \kappa(r)(\mathcal{F}_1(\lambda,\theta',r)\cos\lambda + \mathcal{F}_2(\lambda,\theta',r)\sin\lambda) \quad (2)$$

where $\kappa(r) = (a/r)^2$ and for $j = 1,2$

$$\mathcal{F}_j(\lambda,\theta',r) = \sum_{n=0}^{N-1} \left(\frac{a_e}{r}\right)^{n+1} \sum_{m=0}^{n} (\bar{a}_{n,m}^j \cos m\lambda + \bar{b}_{n,m}^j \sin m\lambda) \bar{P}_{n,m}(\cos\theta') \quad (3)$$

with coefficients

$$\bar{a}_{nm}^j = \left(\frac{a}{a_e}\right)^{n+1} a_{nm}^j, \quad \bar{b}_{nm}^j = \left(\frac{a}{a_e}\right)^{n+1} b_{nm}^j, \quad j = 1,2,$$

and coefficients a_{nm}^j, b_{nm}^j given in Table 2. Note that c_{nm} and s_{nm} from Table 2 are in fact multiplied by $(a/a_e)^n$ in this case, not by $(a/a_e)^{n+1}$ as in the cases of X' and Z'!

If one wants to normalize c_{nm} and s_{nm} from Table 2 with $(a/a_e)^{n+1}$ as in the cases of X' and Z', then (2), (3) can be applied with

$$\kappa(r) = \frac{a_e a}{r^2}, \quad \bar{a}_{nm}^j = \left(\frac{a}{a_e}\right)^{n+2} a_{nm}^j, \quad \bar{b}_{nm}^j = \left(\frac{a}{a_e}\right)^{n+2} b_{nm}^j, \quad j = 1,2.$$

4.4 From spherical harmonic expansions to ellipsoidal harmonic expansions

We further write harmonic expansions $\mathcal{F}(\lambda,\theta',r)$ (and $\mathcal{F}_1, \mathcal{F}_2$) in ellipsoidal-harmonic coordinates (λ,ϕ,u)

$$\begin{cases} r\sin\theta'\cos\lambda &= \sqrt{u^2 + E^2}\sin\phi\cos\lambda, \\ r\sin\theta'\sin\lambda &= \sqrt{u^2 + E^2}\sin\phi\sin\lambda, \\ r\cos\theta' &= u\cos\phi, \end{cases}$$

by mapping the coefficients $c_{nm}, s_{nm}, 0 \le m \le n, 1 \le n \le N$, with Jekeli's transform [1] to $c_{n,m}^{\{ell\}}, s_{n,m}^{\{ell\}}, 0 \le m \le n, 1 \le n \le N_1$:

$$\mathcal{F}(\lambda,\theta',r) = \mathcal{H}(\lambda,\phi,u). \quad (4)$$

The harmonic expansion of $\mathcal{H}(\lambda,\phi,u)$ takes the form

$$\mathcal{H}(\lambda,\phi,u) = \sum_{n=1}^{N_1} \sum_{m=0}^{n} \frac{\bar{S}_{n,m}\left(\frac{u}{E}\right)}{\bar{S}_{n,m}\left(\frac{b}{E}\right)} \left(c_{nm}^{\{ell\}}\cos m\lambda + s_{nm}^{\{ell\}}\sin m\lambda\right) \bar{P}_{nm}(\cos\phi), \quad (5)$$

where $\bar{S}_{n,m}$ are Jekeli's functions and b is the Earth semi-minor axis. In theory $N_1 = \infty$ but in practice $N_1 = N + 40$ (for $N = 740$ or $N = 790$) gives (4) with relative error less than 10^{-20}.

For every fixed u the functions $\mathcal{H}(\lambda, \phi, u)$ and $\mathcal{H}_1(\lambda, \phi, u)\cos\lambda + \mathcal{H}_2(\lambda, \phi, u)\sin\lambda$ are bi-variate trigonometric polynomials of degree N_1 (or N_1+1) and tensor product needlets can be utilized for its fast evaluation (see Subsections 4.6-4.7). The reason for switching from spherical harmonic expansions to ellipsoidal harmonic expansions is to guarantee smaller approximation error.

4.5 Evaluation of magnetic quantities in an ellipsoidal shell

The code `emmsynth_fast.m` computes an approximation $\tilde{G}(\lambda, \phi, u)$ to the magnetic quantity $G(\lambda, \phi, u)$, $G = X', Y', Z'$, for u from the ellipsoidal shell $U_0 \leq u \leq U_1$ with $U_0 = b - 415$ m and $U_1 = b + 1,000,000$ m. This approximation is obtained by interpolating the values of G on several confocal ellipses. For a fixed $(\lambda, \phi) \in \mathbb{S}^2$ if $\tilde{G}(\lambda, \phi, u)$ is the Lagrange interpolant of $G(\lambda, \phi, u)$ at the points $u_j = u_1 + (j-1)h$, $j = 1, 2, \ldots, 2J$, then the error takes the form

$$\tilde{G}(\lambda, \phi, u) - G(\lambda, \phi, u) = \frac{(u - u_1) \cdots (u - u_{2J})}{(2J)!} \frac{\partial^{2J} G}{\partial u^{2J}}(\lambda, \phi, z)$$

for some $z \in (u_1, u_{2J})$. The best choice for u is $u \in [u_J, u_{J+1}]$, where the product $(u - u_1) \ldots (u - u_{2J})$ has a ch^{2J} bound with the smallest constant c. However, the $2J$-th derivative of G with respect to u grows very rapidly as u approaches b (i.e. the Earth surface and below), which leads to big errors!

In order to reduce the influence of the derivative term on the error we take an increasing function μ defined on $[0, \bar{s}]$ such that $\mu(0) = U_0$, $\mu'(0) = 0$, $\mu(\bar{s}) = U_1$ and set

$$g(\lambda, \phi, s) = G(\lambda, \phi, \mu(s)), \quad (\lambda, \phi) \in \mathbb{S}^2, \ 0 \leq s \leq \bar{s}.$$

$$\mu(s) = U_0 + \frac{U_1 - U_0}{\bar{s}^4} s^4.$$

This gives us a function g with *essentially smaller oscillation* of the normal (with respect to the ellipsoids) derivatives than those of G and at the same time these derivatives of g can be explicitly expressed in terms of the normal derivatives of G.

In the code we use $2J$ point Lagrange interpolation for G, where $J = 2$ or $J = 3$. Thus, for appropriate h, setting $s_j = jh$, $j = -J+1, -J+2, \ldots, M+J$ with $\mu(s_{M-1}) < U_1 \leq \mu(s_M)$ we pre-compute the values of $G(\lambda_\ell, \phi_k, v_j)$ at regular points $(\lambda_\ell, \phi_k) \in \mathbb{S}^2$ (see §4.6) on confocal ellipsoids of semi-minor axis $v_j = \mu(s_j)$. Then for the computation of $\tilde{G}(\lambda, \phi, u)$ we utilize this algorithm:

1. For $u \in [U_0, U_1]$ find $s = \mu^{-1}(u)$ and $j \geq 0$ such that $s \in [s_j, s_{j+1}]$, where μ^{-1} denotes the inverse function of μ;

2. Use tensor product needlets (see §4.7) to compute $\tilde{g}(\lambda, \phi, s_i) := \tilde{G}(\lambda, \phi, v_i)$, $i = j-J+1, j-J+2, \ldots, j+J$ from the values of G at the regular points from each of these $2J$ ellipsoids;

3. Use Lagrange interpolation to compute $\tilde{G}(\lambda, \phi, u) := \tilde{g}(\lambda, \phi, s)$ from $\tilde{g}(\lambda, \phi, s_i)$, $i = j - J + 1, j - J + 2, \ldots, j + J$.

Note that the choice of μ as an even function gives $v_{-1} = v_1$, $v_{-2} = v_2$, etc., which allows to only use values of $\tilde{G}(\lambda, \phi, v_j)$ at points with $v_j \geq U_0$ in the computation of $\tilde{G}(\lambda, \phi, u)$! Thus we avoid derivatives of G at points with coordinate u smaller than U_0. The choice $U_0 = b - 415$ m allows to include in the ellipsoidal shell all points *above the surface of the Earth.*

The binary files `emm2015_*.bin` and `emm2017_*.bin` contain the values of $G(\lambda_\ell, \phi_k, v_j)\frac{2}{KL}$, $j = 0, 1, \ldots, M + J$, for $G = X', Y', Z'$ or their seqular variations. In order to cover the whole ellipsoidal shell $[U_0, U_1]$ we use in our code 45 ellipsoids for $J = 2$ (i.e. $M + J = 45$) or 30 ellipsoids for $J = 3$. For K and L see formula (6) and Subsection 4.6.

4.6   Regular grids

The regular grids on ellipsoids we use in the codes are given in ellipsoidal-harmonic coordinates (λ, ϕ) by

$$\mathcal{X} = \{x_{k,\ell} = (\lambda_\ell, \phi_k) \; : \; k = 0, 1, \ldots, K, \; \ell = 0, 1, \ldots, L - 1\},$$

where

$$\phi_k = \frac{\pi k}{K}, \; k = 0, 1, \ldots, K; \qquad \lambda_\ell = \frac{2\pi\ell}{L}, \; \ell = 0, 1, \ldots, L - 1.$$

Here L *must be even* in order to allow the values of the same grid to be used for continuation through the poles.

In our code $K = 1481$ and $L = 2960$ for both magnetic models.

4.7   Evaluation of bi-variate trigonometric polynomials

As shown in §4.3–4.4 the restriction of any of the quantities $f = X'/\kappa$, Y'/κ, Z'/κ on any ellipsoid confocal with the Earth reference ellipsoid is a bi-variate trigonometric polynomial in λ and ϕ of degree at most $N_1 + 1$. Similarly, the restrictions of f on any sphere centred at the centre of the Earth is a bi-variate trigonometric polynomial in λ and θ' of degree at most N.

The bi-variate trigonometric polynomial $f(\lambda, \phi)$ is evaluated at a point $(\lambda, \phi) \in \mathbb{S}^2$ by tensor product trigonometric needlets of the form

$$\tilde{f}(\lambda, \phi) = \sum_{|\phi - \phi_k| \leq \delta_1} \sum_{|\lambda - \lambda_\ell| \leq \delta_2} \mathcal{K}_1(\phi - \phi_k)\mathcal{K}_2(\lambda - \lambda_\ell)\frac{2}{KL}f(\lambda_\ell, \phi_k), \qquad (6)$$

where $\mathcal{K}_1$ and $\mathcal{K}_2$ are trigonometric needlet kernels in λ and ϕ, respectively.

We chose in the code both the number of knots $\{\phi_k\}$ on the interval $[\phi - \delta_1, \phi + \delta_1]$ and the number of knots $\{\lambda_\ell\}$ on the interval $[\lambda - \delta_2, \lambda + \delta_2]$ to be 18 for target accuracy `ParKer(3)`$= 0.43 \cdot 10^{-5}$.

In (6) for λ close to 0 or to 2π we assume that the definition of λ_ℓ from §4.6 is extended by the same formula for $\ell < 0$ or $\ell \geq L$, which implies the *periodic* extension $f(\lambda + 2\pi, \phi) = f(\lambda, \phi)$ of f. Similarly, for ϕ close to 0 or to π we extend the definition of ϕ_k from §4.6 by the same formula for $k < 0$ or $k > K$, which implies the *even semi-periodic* extension $f(\lambda + \pi, -\phi) = f(\lambda, \phi)$ in the case of Z', and the *odd semi-periodic* extension $f(\lambda + \pi, -\phi) = -f(\lambda, \phi)$ in the case of X', Y'. These extensions do not require evaluation of the polynomial f at new points (λ_ℓ, ϕ_k) whenever L is even!

The extended grid dimensions in our code are 1498×2977 nodes.

4.8   Accuracy

For every component G of the magnetic field, where G denotes one of $X', Y', Z', X, Y, Z, H, F$ as given by the magnetic model, our code is designed to compute an approximation $\tilde{G}$ to G so that

$$|\tilde{G}(\lambda, \phi, h, t) - G(\lambda, \phi, h, t)| < 1 \ nT,$$

where $-180^0 \leq \lambda \leq 180^0$, $-90^0 \leq \phi \leq 90^0$, $-415 \leq h \leq 1,000,000$ m and $2015 \leq t \leq 2020$ for EMM2015 or $2017 \leq t \leq 2022$ for the predictive part of EMM2017.

The largest error we have observed for any of these quantities does not exceed 0.3251 nT for EMM2015 or 0.4804 nT for the predictive part of EMM2017.

5 Code `emmsynth_fast.m`

The code `emmsynth_fast.m` is the main program designed to perform fast an accurate evaluation of the geomagnetic field components. The code uses:

- the keyboard input of the selected magnetic model (enter 1 for EMM2015 or 2 for the predictive part of EMM2017);

- the keyboard input of the decimal year of computation in the range from 2015 to 2020 for EMM2015 or in the range from 2017 to 2022 for EMM2017;

- the `emmsynth_init.m` produced six binary data files `emm2015_*.bin` or `emm2017_*.bin` (depending on the selected magnetic model) as described in §2.4;

- the user-defined point coordinates file `scattered_points.dat` as described in §2.1.

The output is given in the file `scattered_points_values.dat`. It contains the magnetic field components' values, as described in §2.2, at the scattered points from the input file `scattered_points.dat`.

The algorithm for approximate evaluation of the magnetic field is described in §4. For the fast evaluation of the magnetic field at 15,000,000 points MATLAB uses 8.24 GB of RAM, while for 30,000,000 points it needs 10.83 GB. The code can be easily modified to parallel the input, the approximate evaluation and the output in order to reduce the memory usage.

6 Code `emmsynth_init.m`

The code `emmsynth_init.m` is an auxiliary program designed to produce the six binary data files `emm201?_*.bin`, which are not included in the package because of their 9 GB size in total for the respective model. These files are necessary to initialize `emmsynth_fast.m`.

The user selects from the keyboard the magnetic model (enter 1 for EMM2015 or 2 for the predictive part of EMM2017). The code computes the magnetic field components' weighted values at regular points located on confocal ellipsoids from the EMM2015 coefficients files `EMM2015.COF` and `EMM2015SV.COF` or from the EMM2017 coefficients files `EMM2017.COF` and `EMM2017SV.COF`. The "standard" method for evaluating *surface* spherical harmonic *gridded* values is applied. The grid is further extended as explained in §4.6 and §4.7.

On the computer where the tests were performed it takes approximately 17.3 minutes to run the code for EMM2015 and 18.5 minutes for EMM2017. The memory requirements for `emmsynth_init.m` are weaker than the memory requirements for `emmsynth_fast.m`.

7 Code `emmsynth_standard.m`

The purpose of the code `emmsynth_standard.m` is to test the *accuracy* of the main code `emmsynth_fast.m` and to be used as a benchmark for its *speed*.

The code `emmsynth_standard.m` evaluates the magnetic field components using the standard methods for spherical harmonic computation based of the model coefficients included in `EMM201?.COF` and `EMM201?SV.COF`. Note that both the north component in geocentric coordinates X' and the east component in geocentric coordinates Y' are represented as sums of two harmonic functions in order to achieve stable representations everywhere on the Earth. The evaluation of the spherical harmonic series is done in spherical coordinates without passing to ellipsoidal harmonic coordinates as in the other two codes.

The input for the testing program consists of the selected magnetic model and the decimal year of computation (as for `emmsynth_fast.m`), the magnetic model coefficients read from `EMM201?.COF` and `EMM201?SV.COF` and the file `scattered_points_values.dat`. In order to check the accuracy of `emmsynth_fast.m` one runs `emmsynth_standard.m` with the same magnetic model and year on the `emmsynth_fast.m` output `scattered_points_values.dat`.

Among the displayed program statistics of `emmsynth_standard.m` one can find the absolute errors of the magnetic field components computed by `emmsynth_fast.m`. The speed of our realization of the two methods can be compared using the reported numbers "Retrieved points per second". In both programs these numbers are formed on the base of pure computational time, ignoring the time necessary to read the input or to write the output.

When `emmsynth_standard.m` is run after `emmsynth_fast.m` one should use the input file `scattered_points.dat` with no more than 60,000 points due to the slow speed of the "standard" code. (On the standard laptop we use (see §8) `emmsynth_standard.m` processes 60,000 points for more than 10 minutes.) The memory requirements for `emmsynth_standard.m` are weaker than the memory requirements for `emmsynth_fast.m`.

8 Experiments

The software has been extensively tested on a PC with Intel Core i7-8759H CPU @ 2.2 GHz, 32 GB of RAM and 1000 GB SSD used to store the binary data files emm201?_*.bin and the input file scattered_points.dat. As reported in §5 program emmsynth_fast.m uses less than 11 GB of RAM for retrieving 30,000,000 points.

The tests are conducted using MATLAB R2016a under WINDOWS 10.

8.1 The test files

The test input file scattered_points.dat consists of the geographical geodetic coordinates of 1,000 (or 1,000,000) points, such that the latitude, the longitude and height are randomly distributed in the range $[-90, 90]$, $[-180, 180)$ degrees and $[-0.415, 1000]$ km, respectively.

8.2 Tests

Testing statistics of program emmsynth_init.m, program emmsynth_fast.m for the test input files scattered_points.dat with 1,000,000 and 1,000 points and of program emmsynth_standard.m with 1,000 points follow.

8.2.1 Tests with EMM2015

```
-----------------------------------------------------------------
  Model: EMM2015,        Program:  emmsynth_init
-----------------------------------------------------------------
 Total time                   = 1037.083 CPU seconds
 Coefficients load time       = 1.047 CPU seconds
 Coefficients formation time  = 9.184 CPU seconds
 Synthesis time               = 1020.297 CPU seconds
 Values write time            = 2.831 CPU seconds
-----------------------------------------------------------------


-----------------------------------------------------------------
      Program:  emmsynth_fast,       Model: EMM2015
  Evaluation year:             = 2018.75
-----------------------------------------------------------------
  Total time                   = 75.163 CPU seconds
  Grid values load time        = 8.204 CPU seconds
  Coordinates load time        = 2.046 CPU seconds
  Synthesis time               = 56.728 CPU seconds
  Values write time            = 7.842 CPU seconds
  Total number of points       = 1000000
  Retrieved points per second  = 17627.9791
-----------------------------------------------------------------
```

```
----------------------------------------------------------------
        Program:  emmsynth_fast,       Model: EMM2015
     Evaluation year:              = 2018.75
----------------------------------------------------------------
     Total time                 = 11.7 CPU seconds
     Grid values load time      = 11.278 CPU seconds
     Coordinates load time      = 0.015 CPU seconds
     Synthesis time             = 0.063 CPU seconds
     Values write time          = 0.015 CPU seconds
     Total number of points     = 1000
     Retrieved points per second = 15873.0159
----------------------------------------------------------------


----------------------------------------------------------------
        Program:  emmsynth_standard,      Model: EMM2015
     Evaluation year:              = 2018.75
----------------------------------------------------------------
     Total time                 = 11.137 CPU seconds
     Coordinate load time       = 0.014 CPU seconds
     Model load time            = 0.313 CPU seconds
     Coefficient formation time = 0.078 CPU seconds
     Synthesis time             = 10.732 CPU seconds
     Values write time          = 0 CPU seconds
     Total number of points     = 1000
     Execution time per point   = 0.010732 CPU seconds
     Retrieved points per second = 93.1793
----------------------------------------------------------------
     Maximal absolute errors at the given scattered points
----------------------------------------------------------------
Xprime:               0.019147
Yprime:               0.0090415
Zprime:               0.030893
North component X:     0.019133
East component Y:      0.0090415
Down component Z:      0.030803
Horizontal intensity H: 0.015265
Total intensity F:     0.032836
Inclination I:         6.5538e-07
Declination D          3.9706e-06
----------------------------------------------------------------
```

8.2.2 Tests with EMM2017

```
-----------------------------------------------------------------
       Program:  emmsynth_init,       Model: EMM2017
-----------------------------------------------------------------
 Total time                  = 1109.383 CPU seconds
 Coefficients load time      = 1.078 CPU seconds
 Coefficients formation time = 10.493 CPU seconds
 Synthesis time              = 1092.383 CPU seconds
 Values write time           = 2.828 CPU seconds
-----------------------------------------------------------------


-----------------------------------------------------------------
       Program:  emmsynth_fast,       Model: EMM2017
 Evaluation year:            = 2018.75
-----------------------------------------------------------------
 Total time                  = 74.889 CPU seconds
 Grid values load time       = 8.53 CPU seconds
 Coordinates load time       = 2.046 CPU seconds
 Synthesis time              = 56.112 CPU seconds
 Values write time           = 7.842 CPU seconds
 Total number of points      = 1000000
 Retrieved points per second = 17821.4999
-----------------------------------------------------------------


-----------------------------------------------------------------
       Program:  emmsynth_fast,       Model: EMM2017
 Evaluation year:            = 2018.75
-----------------------------------------------------------------
 Total time                  = 8.877 CPU seconds
 Grid values load time       = 8.377 CPU seconds
 Coordinates load time       = 0 CPU seconds
 Synthesis time              = 0.062 CPU seconds
 Values write time           = 0.032 CPU seconds
 Total number of points      = 1000
 Retrieved points per second = 16129.0323
-----------------------------------------------------------------


-----------------------------------------------------------------
       Program:  emmsynth_standard,       Model: EMM2017
 Evaluation year:            = 2018.75
-----------------------------------------------------------------
 Total time                  = 13.027 CPU seconds
 Coordinate load time        = 0.014 CPU seconds
 Model load time             = 0.328 CPU seconds
 Coefficient formation time  = 0.109 CPU seconds
 Synthesis time              = 12.56 CPU seconds
```

```
    Values write time          = 0.016 CPU seconds
    Total number of points     = 1000
    Execution time per point   = 0.01256 CPU seconds
    Retrieved points per second = 79.6178
--------------------------------------------------------------
     Maximal absolute errors at the given scattered points
--------------------------------------------------------------
Xprime:                0.015314
Yprime:                0.012529
Zprime:                0.036069
North component X:     0.015332
East component Y:      0.012529
Down component Z:      0.036012
Horizontal intensity H: 0.015301
Total intensity F:     0.037703
Inclination I:         2.2742e-07
Declination D          5.3011e-06
--------------------------------------------------------------
```

8.2.3 Test Comments

The absolute errors measured for the 1,000 point test file vary from 0.032836 to 0.0090415 nT for EMM2015 and from 0.037703 to 0.012529 nT for EMM2017. The largest absolute errors we have observed are 0.3251 nT for EMM2015 and 0.4804 nT for EMM2017. Recall that the code is designed to evaluate the magnetic field elements with error not exceeding 1 nT.

The improvement in computational speed of our code (measured by `Retrieved points per second`) compared to the software using the standard spherical harmonic series method is given in the last column of Table 3.

Enhanced Magnetic Model	emmsynth_fast (1,000,000 points)	emmsynth_standard (1,000 points)	improvement (times)
EMM2015	17627.98	93.18	189.18
EMM2017	17821.50	79.62	223.83

Table 3: `Retrieved points per second` for `emmsynth_standard.m` and `emmsynth_fast.m` and the improvement in computational speed

The 1,000,000 point input file is used to measure the speed of `emmsynth_fast.m` because the speed for the 1,000 point input file is unreliable due to the small CPU time interval. The computational speed of `emmsynth_standard.m` is stable for the 1,000 point input file.

The larger improvement in speed for EMM2017 is induced by the slower speed of `emmsynth_standard.m` for this model, which is caused by its higher degree – 790 for EMM2017 compared with 740 for EMM2015.

8.3 Other experiments

The code `emmsynth_fast.m` has been tested on up to $30,000,000$ points, which were processed at the speed of the second column of Table 3.

The code `emmsynth_fast.m` uses six data files `emm201?_*.dat` of total size 9 GB. We have modified `emmsynth_init.m` and `emmsynth_fast.m` to `emmsynth_init0.m` and `emmsynth_fast0.m`, which produced and used data files of total size 6 GB. In order to achieve the same error bound of 1 nT on smaller amount of data the code `emmsynth_fast0.m` uses 5-th degree Lagrange interpolation (cf §4.5), which reduces the speed, as shown in Table 4.

Enhanced Magnetic Model	emmsynth_fast0 (1,000,000 points)	emmsynth_standard (1,000 points)	improvement (times)
EMM2015	11911.99	93.18	127.84

Table 4: `Retrieved points per second` for `emmsynth_standard.m` and `emmsynth_fast0.m` and the improvement in computational speed

9   Conclusions

The experiments with the software described above demonstrate the capability of our needlet method for fast evaluation of magnetic field components represented in terms of *solid* spherical harmonics at scattered points *in space*. The current version `emmsynth_fast.m` of our software runs from 189 to 224 times faster than the software using the standard spherical harmonic series method.

References

[1] C. Jekeli: The exact transformation between ellipsoidal and spherical harmonic expansions. *Manuscripta Geodaetica*, **13** (1988), 106-113.

[2] K. G. Ivanov, N. K. Pavlis, P. Petrushev: Precise and Efficient Evaluation of Gravimetric Quantities at Arbitrarily Scattered Points in Space *Journal of Geodesy*, **92** (2018), 779–796.
https://doi.org/10.1007/s00190-017-1094-y