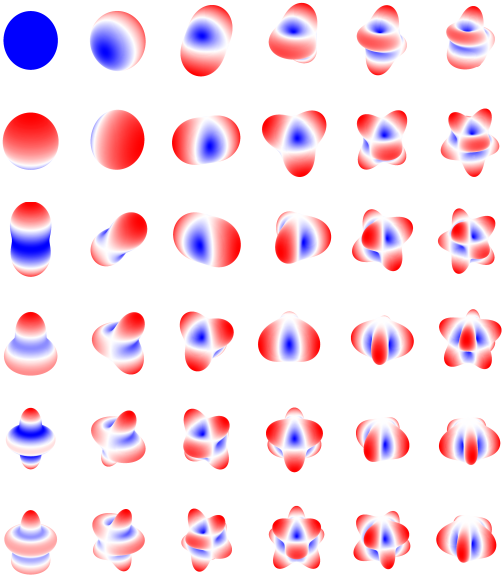
10 Years with Spherical Harmonics

Blažej Bucha

Department of Theoretical Geodesy and Geoinformatics
Slovak University of Technology in Bratislava
blazej.bucha@stuba.sk

Tatry 2022: Globálna geodézia a geoinformatika

What are Spherical Harmonics Anyway?



What are Spherical Harmonics Anyway?

(Almost) Any function on a sphere can be computed using spherical harmonics.

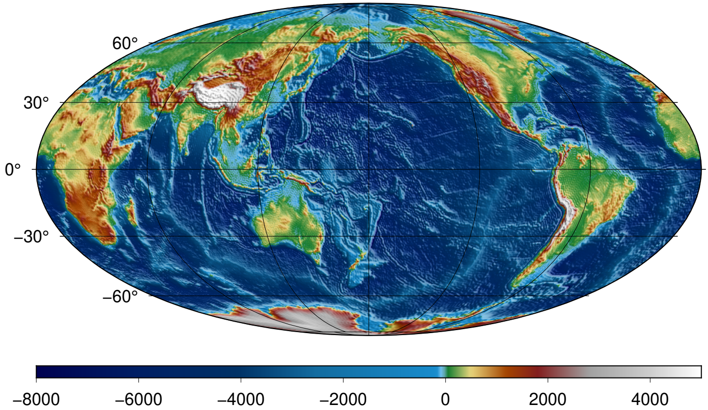


Figure: Earth's topography and bathymetry (m) expanded up to spherical harmonic degree 360.

What are Spherical Harmonics Anyway?

(Almost) Any function on a sphere can be computed using spherical harmonics.

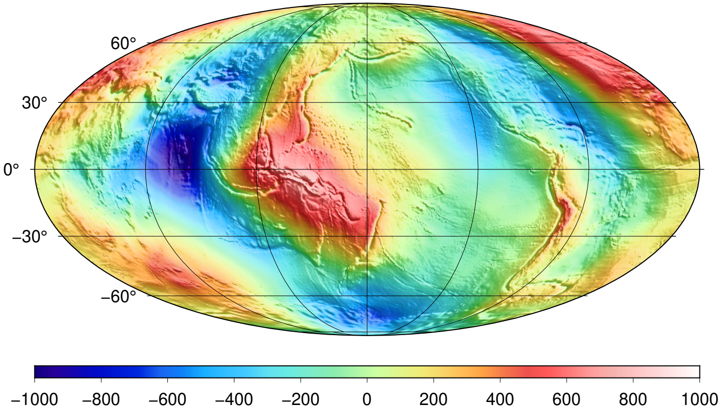


Figure: Disturbing potential (m^2 s^{-2}) on the GRS80 ellipsoid expanded up to degree 720.

What are Spherical Harmonics Anyway?

(Almost) Any function on a sphere can be computed using spherical harmonics.

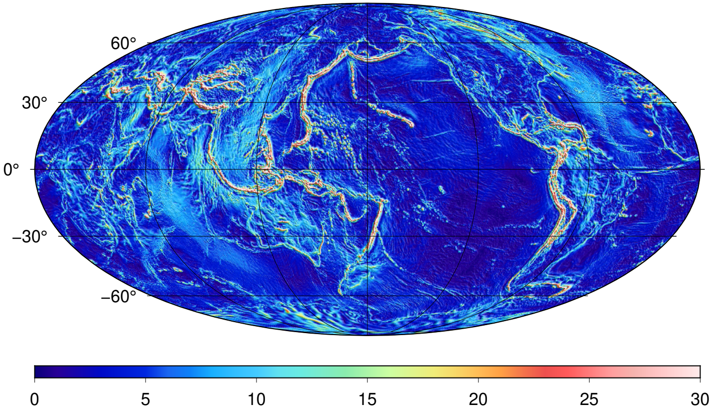


Figure: Total deflection of the vertical (arcsec) on the GRS80 ellipsoid expanded up to degree 720.

The Naive Way (2011)

Goal: Compute the following equation at dense grids (φ_i, λ_j) with many spherical harmonics as efficiently as possible:

$$f(\varphi_i, \lambda_j) = \sum_{n=0}^{n_{\max}} \sum_{m=0}^{n} \left(\bar{C}_{nm} \cos k\lambda_j + \bar{S}_{nm} \sin k\lambda_j\right) \bar{P}_{nm}(\sin\varphi_i) \qquad (1)$$

For $n_{\max} = 1000$, there is ∼1,000,000 spherical harmonics.
For $n_{\max} = 10{,}000$, there is ∼100,000,000 spherical harmonics.

The Naive Approach: Simply compute all the terms in Eq. (1) for all the grid points (φ_i, λ_j) and do the summation.

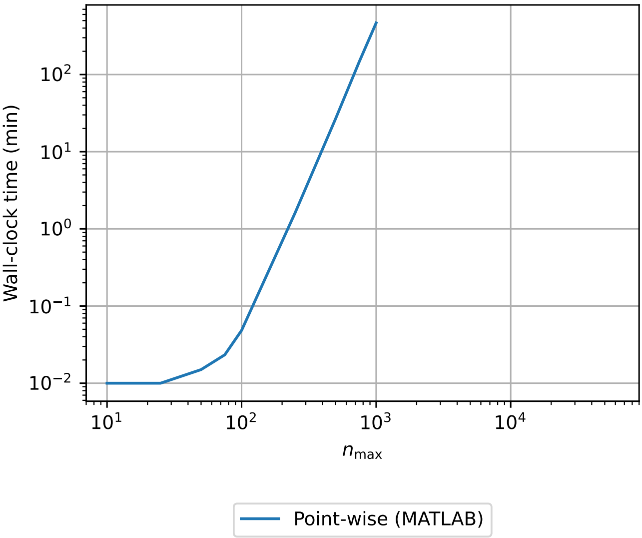


Figure: Computation time as a function of maximum harmonic degree in a log-log scale (GrafLab, Point-Wise mode)

Lumped coefficients (2012)

After re-ordering the two summations, we get

$$\begin{aligned} f(\varphi_i, \lambda_j) &= \sum_{m=0}^{n_{\max}} \sum_{n=0}^{n_{\max}} \left(\bar{C}_{nm} \, \cos k\lambda_j + \bar{S}_{nm} \, \sin k\lambda_j\right) \bar{P}_{nm}(\sin\varphi_i) \\ &= \sum_{m=0}^{n_{\max}} A_m(\varphi_i) \, \cos k\lambda_j + B_m(\varphi_i) \, \sin k\lambda_j, \end{aligned} \tag{2}$$

where

$$\begin{aligned} A_m(\varphi_i) &= \sum_{n=0}^{n_{\max}} \bar{C}_{nm} \, \bar{P}_{nm}(\sin\varphi_i) \\ B_m(\varphi_i) &= \sum_{n=0}^{n_{\max}} \bar{S}_{nm} \, \bar{P}_{nm}(\sin\varphi_i) \end{aligned} \tag{3}$$

are lumped coefficients that are constant for a fixed φ_i. Eq. (2) can be computed using FFT.

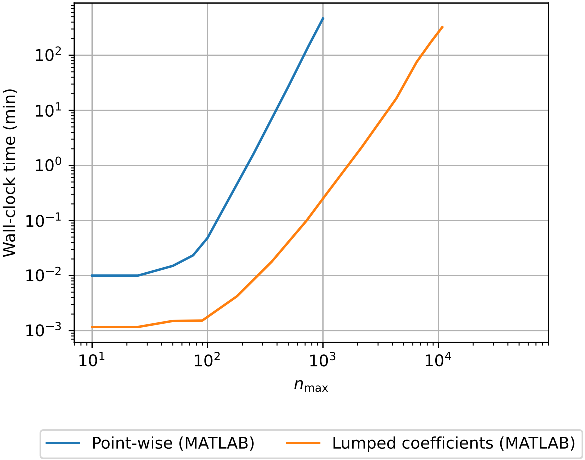


Figure: Computation time as a function of maximum harmonic degree in a log-log scale (GrafLab, Grid-Wise mode)

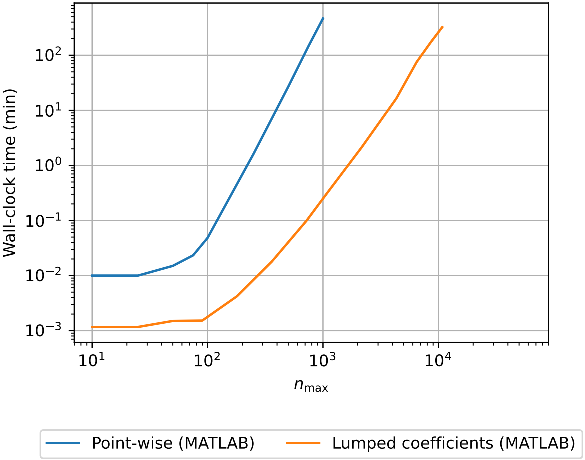


Figure: Computation time as a function of maximum harmonic degree in a log-log scale (GrafLab, Grid-Wise mode)

Speed up factor up to ∼1500!

The Equatorial Symmetry (2018)

Equatorial symmetry of Legendre functions:

$$\bar{P}_{nm}(\sin(-\varphi)) = (-1)^{n+m}\,\bar{P}_{nm}(\sin\varphi). \tag{4}$$

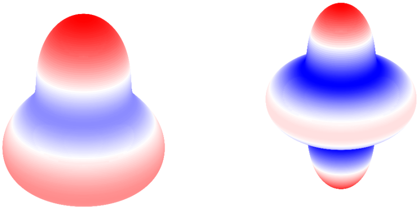


Figure: *Left*: $Y_{30}(\varphi, \lambda)$, *right*: $Y_{40}(\varphi, \lambda)$

The C Language (2019)

- Low-level compiled general-purpose programming language
- Highly portable
- Used from embedded systems to supercomputers

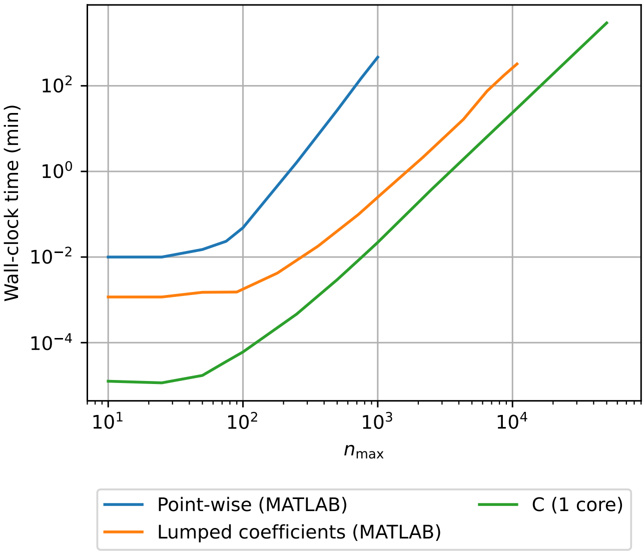


Figure: Computation time as a function of maximum harmonic degree in a log-log scale (C language, 1 core)

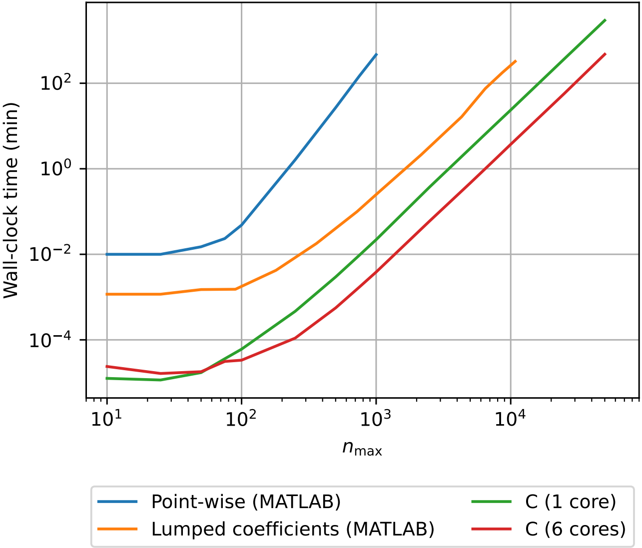
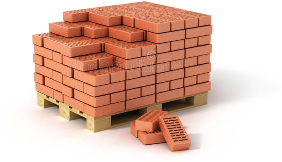
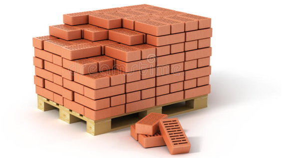


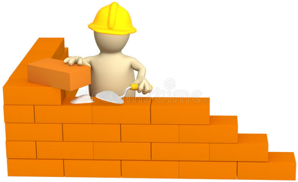
Figure: Computation time as a function of maximum harmonic degree in a log-log scale (C language, 6 cores)

CPU Caching (2020)

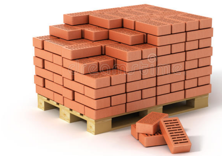


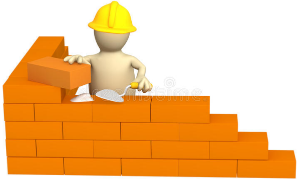
CPU Caching (2020)





CPU Caching (2020)





Vector CPU instructions in C (2022)

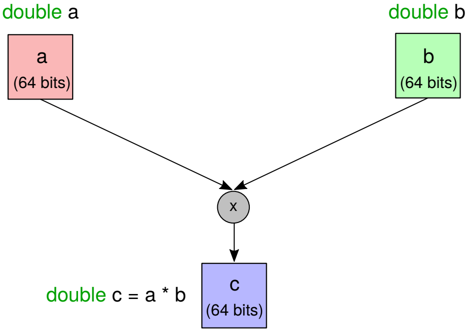


Figure: Scalar multiplication of two doubles

Vector CPU instructions in C (2022)

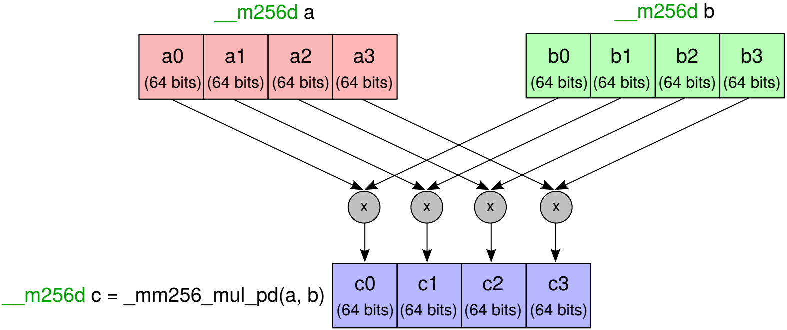


Figure: Multiplication of two vectors of doubles using AVX2

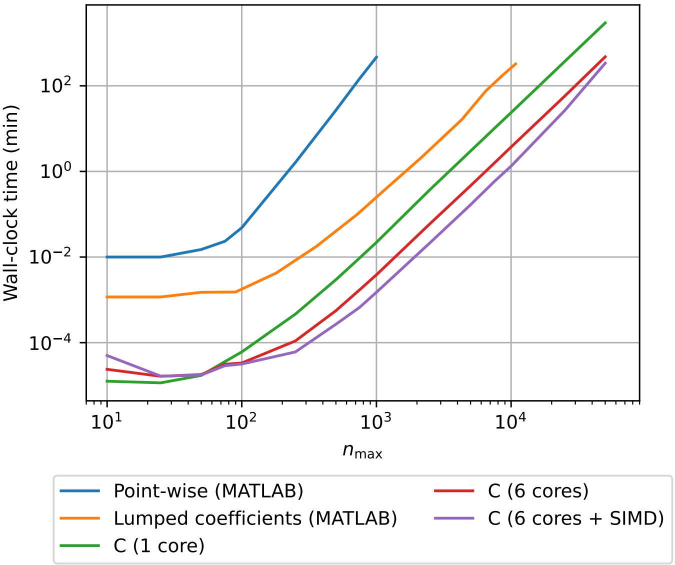


Figure: Computation time as a function of maximum harmonic degree in a log-log scale (C language, 6 cores, SIMD CPU instructions)

Future Work (2022 – ???)

CHarm: C library to work with spherical harmonics up to almost arbitrary degrees

- `https://github.com/blazej-bucha/charm`

Future work:

- Other normalization schemes
- MPI parallelization for distributed-memory systems
- Polar optimization
- Object-oriented Python wrapper with ctypes (in progress)
- Fused multiply-accumulate CPU instruction
- Build CHarm with CMake on Windows?

Thank you for your attention!

Backup slides

CPU Caching (2020)

RAM

CPU Caching (2020)

RAM

Bus

CPU Caching (2020)

RAM

Bus

CPU

CPU Caching (2020)

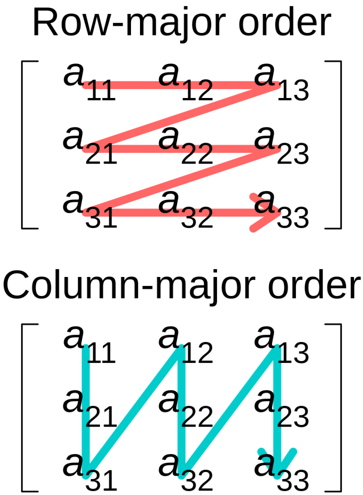


Figure: Memory storage schemes (*source*: `https://www.wikipedia.org`). *Top*: the C language, *bottom*: Fortran

CPU Caching (2020)

Cache-friendly code in C

```
for (size_t i = 0; i < N; i++)
{
    for (size_t j = 0; j < N; j++)
    {
        c += A[i][j];
    }
}
```

Cache-friendly code in Fortran

```
do i = 1, N
    do j = 1, N
        c = c + A(j, i)
    end do
end do
```

Vector CPU instructions in C (2022)

- Low-level programming
  - Assembly language
  - C intrisic functions from the `immintrin.h` header file
- Requires specific data alignement (`malloc` is not suitable)
- Often requires completely new code and algorithms
- Quadruple precision not supported on the hardware level

Table: Overview of AVX instruction sets

Instruction sets	Register size (bits)	Single precision (float)	Double precision (double)	Introduced (year)
AVX	128	8	4	2011
AVX2	256	8	4	2013
AVX-512	512	16	8	2016