

مقایسه روش‌های سطری و ستونی تبدیل هماهنگ کروی از درجه/مرتبه خیلی بالا

مهدی گلی^{۱*}

^۱ استادیار، دانشکده مهندسی عمران دانشگاه صنعتی شاهرود، شاهرود، ایران

(دریافت: ۱۳۹۹/۰۵/۰۹، پذیرش: ۱۳۹۹/۱۰/۰۲)

چکیده

تبدیل هماهنگ کروی، ابزار استاندارد را برای تحلیل/ساخت طیفی داده‌های ژئودتیکی و ژئوفیزیکی نظیر میدان ثقل و میدان مغناطیسی زمین، توپوگرافی و جو فراهم می‌آورد. به لطف مأموریت‌های اخیر فضایی نظیر SRTM، تفکیک مکانی داده‌های زمین مرجع نظیر توپوگرافی، پیوسته رو به کاهش است که این موضوع، افزایش درجه/مرتبه مدل هماهنگ را در پی دارد. محاسبه توابع وابسته لژاندر در درجات بالا ($n > 2000$) با مشکل عددی خطای پاریز روبه‌رو می‌شود. برای رفع این مشکل، روش‌های مختلف سطری و ستونی محاسبه توابع وابسته لژاندر در مطالعات پیشین پیشنهاد شده است. هدف از مطالعه حاضر، بررسی کارایی این روش‌ها در تبدیل تحلیل و ساخت هماهنگ کروی است. الگوریتم سطری و ستونی تحلیل و ساخت هماهنگ کروی در یک نرم‌افزار به زبان برنامه‌نویسی فرترن و با توان پردازش موازی اجرا شده است. برای محاسبه توابع وابسته لژاندر در حالت سطری از روش بلیکوف (Belikov) و در حالت ستونی از روش حساب برد میسوا (ERA) استفاده شد. نتایج عددی مبتنی بر نرم-افزار توسعه داده شده نشان می‌دهد هر دو روش سطری و ستونی در بازه درجه/مرتبه ۲۱۶۰ تا ۲۱۶۰۰، دقت یکسانی در تبدیل هماهنگ کروی دارند. همچنین سرعت روش سطری نسبت به ستونی در پردازش سری و موازی به ترتیب ۱/۸ و ۱/۲ برابر سریع‌تر است.

واژه‌های کلیدی: توابع وابسته لژاندر، تبدیل هماهنگ کروی، روش بلیکوف، روش ERA

۱ مقدمه

بسط سری هماهنگ کروی تابع کراندار $f(\theta, \lambda)$ با رابطه زیر تعریف می‌شود (هایسکانن و موریتز، ۱۹۶۷):

$$f(\theta, \lambda) = \sum_{n=0}^N \sum_{m=0}^n C_{nm} Y_{nm}^c(\theta, \lambda) + S_{nm} Y_{nm}^s(\theta, \lambda) \quad (1)$$

که (θ, λ) مختصات کروی، f_{nm} ضرایب سری هماهنگ‌های کروی و Y_{nm}^c و Y_{nm}^s هماهنگ‌های کروی از درجه n و مرتبه m هستند:

$$\begin{cases} Y_{nm}^c(\theta, \lambda) \\ Y_{nm}^s(\theta, \lambda) \end{cases} = P_{nm}(\cos \theta) \begin{cases} \cos m\lambda \\ \sin m\lambda \end{cases} \quad (2)$$

P_{nm} توابع وابسته لژاندر (نوع اول) هستند. توابع وابسته لژاندر، توابع ویژه معادله دیفرانسیل

$$\frac{d}{dt}(1-t^2) \frac{d}{dx} P_{nm}(x) \left(n(n+1) - \frac{m^2}{1-t^2} \right) P_{nm}(t) = 0 \quad (3)$$

هستند که در این رابطه $t = \cos \theta$ است. توابع کاملاً نرمال‌شده وابسته لژاندر (fnALF) عبارت‌اند از (هایسکانن و موریتز، ۱۹۶۷):

$$\bar{P}_{nm}(t) = \begin{cases} \sqrt{(2-\delta_{m,0})(2n+1)} \frac{(n-m)!}{(n+m)!} P_{nm}(t) & 0 \leq m \leq n \\ 0 & m > n \end{cases} \quad (4)$$

که $\delta_{m,0}$ دلتای کروکر است. بر اساس قضیه نمونه‌برداری روی کره، درجه سری هماهنگ رابطه (۱) به نرخ نمونه‌برداری داده‌ها وابسته است. بیشینه درجه بازیابی‌شونده برای داده‌ها با فاصله کمانی منظم Δ برابر $N = \frac{\pi}{\Delta}$ است (اسنیو، ۱۹۹۴). در این مطالعه تنها هماهنگ‌های کروی سطحی و توابع تعریف‌شونده روی سطح کره/بیضوی بررسی می‌شوند. بدیهی است برای توابع سه‌بعدی که تابعی از (r, θ, λ) هستند، به هماهنگ‌های کروی فضایی نیاز است، اما با توجه به رابطه

هماهنگ‌های کروی سطحی و فضایی، تمامی مطالب این مطالعه درباره تحلیل هماهنگ کروی فضایی توابع سه‌بعدی نظیر میدان پتانسیل نیز صادق است.

به لطف فناوری‌های نوین ماهواره‌ای نظیر سنجش از دور، سامانه‌های راداری و تعیین موقعیت ماهواره‌ای، تفکیک مکانی داده‌های زمین مرجع کاهش چشمگیری داشته است؛ برای مثال در مدل توپوگرافی SRTM15+، ارتفاعات سراسر زمین با تفکیک مکانی ۱۵ ثانیه درجه‌ای (معادل حدود ۵۰۰ متر روی استوا) در دسترس است (تزر و همکاران، ۲۰۱۹). در مطالعه رکسر و هرت (۲۰۱۵) از نسخه اولیه این مدل برای تولید مدل هماهنگ توپوگرافی زمین تا درجه/مرتبه ۴۳۲۰۰ استفاده شده است.

بخش عمده محاسبات تبدیل هماهنگ کروی (SHT) معطوف به نحوه محاسبه fnALF است. در اینجا منظور از SHT، تحلیل هماهنگ کروی (SHA) و ساخت هماهنگ‌های کروی (SHS) است (جکلی، ۲۰۱۷). این مسئله با افزایش درجه هماهنگ N اهمیت بیشتری پیدا می‌کند؛ زیرا با افزایش N در محاسبه fnALF خطای عددی نظیر پاریز یا ناپایداری رخ می‌دهد و بنابراین محاسبه fnALF مستلزم استفاده از روش‌هایی اضافی جهت غلبه بر این خطاهاست. استفاده از این روش‌ها اغلب باعث کاهش سرعت محاسبات حتی تا ۵۰ برابر حالت استاندارد می‌شود (ویتور و همکاران، ۲۰۰۸). به کارگیری یک روش مناسب با کمترین اثر در کاهش سرعت و دقت محاسبات، باعث افزایش کارایی تبدیل هماهنگ کروی می‌شود.

در مطالعات پیشین، روش‌های متعددی برای محاسبه سریع و دقیق fnALF برای درجه و مرتبه‌های بالا معرفی شده است. بیشتر این مطالعات به مقایسه کارایی روش‌های مختلف fnALF با یکدیگر پرداخته‌اند. این در حالی است که محاسبه fnALF به تنهایی، اهمیتی در بحث SHT ندارد. از آنجاکه الگوریتم محاسبه عددی SHA و SHS به

برای حل مشکل خطای پاریز، روش‌های مختلفی پیشنهاد شده است که در بین آنها استفاده از ایده حساب برد مبسوط ((Extended Range Arithmetic, ERA) نسبت به سایرین از سرعت بیشتری برخوردار است (انوموتو، ۲۰۱۵ و نوبکوا و دمیترنکو، ۲۰۱۶). در این روش، هر عدد غیر صفر X را می‌توان به صورت حاصل ضرب یک عدد اعشاری x و یک متغیر صحیح iX نوشت:

$$X = xB^{iX}$$

که B یک عدد توانی تا حد امکان بزرگ در پایه ۲ است. در دقت مضاعف، چنانچه عددی در بازه B و $\frac{1}{B}$ قرار بگیرد، $iX = 0$ است، اما به طور کلی مقدار iX برابر $\text{int}\left(\frac{X}{B}\right)$ است. براساس برآورد فوکوشیما (۲۰۱۲)، استفاده از ERA باعث کاهش سرعت محاسبات به میزان حدود ۳۰٪ نسبت به روش استاندارد (رابطه ۵) می‌شود. ایده استفاده از ERA برای محاسبه fnALF را پرزو و رینکه (۲۰۱۰)، نسوآدا (۲۰۱۱) و فوکوشیما (۲۰۱۲) GrafLab.m در نرم‌افزارهای ERA (بوچا و یانک، ۲۰۱۳) و SHbundle دانشگاه اشتوتگارت نیز استفاده شده است. علاوه بر این، رکسر و هرت (۲۰۱۵) نرم‌افزار ERA مطالعه فوکوشیما (۲۰۱۲) را در نرم‌افزار SHTOOLS اجرا کردند و توانستند مدل توپوگرافی زمین را تا درجه/مرتبه ۴۳۲۰۰ تحلیل کنند.

بلیکوف (۱۹۹۱) روش تکراری چهار نقطه‌ای را جایگزین رابطه (۶) کرد. این روش توانایی محاسبه fnALF را تا درجه/مرتبه خیلی بالا بدون خطای عددی پاریز دارد. معادلات این روش عبارت‌اند از:

$$P_{n,0} = a_n x P_{n-1,0} - \frac{u}{2} b_n P_{n-1,1}, \quad n > 0, m = 0$$

$$a_n = \sqrt{\frac{2n+1}{2n-1}}, \quad b_n = a_n \sqrt{\frac{2(n-1)}{n}}$$

$$P_{n,m} = c_{nm} t P_{n-1,m} - d_{nm} u P_{n-1,m+1} + e_{nm} u P_{n-1,m-1}, \quad n > 0, m > 0$$

$$c_{nm} = \frac{a_n}{n} \sqrt{(n+m)(n-m)}$$

نحوه محاسبه fnALF وابسته است، باید اثر این روش‌ها در کارایی SHT دیده شود درحالی‌که این بخش در مطالعات قبلی در نظر گرفته نشده و بررسی آن هدف اصلی این مقاله است.

۲ مبانی نظری

برای محاسبه fnALF معمولاً از روش‌های تکراری استفاده می‌شود. روش متداول، روابط سه نقطه‌ای پیشرو ستونی است که در اغلب مطالعات نظیر هلمز و فیدرسون (۲۰۰۲)، فوکوشیما (۲۰۱۲) و بالمینو (۲۰۱۲) پیشنهاد و در نرم‌افزارهای مختلف نظیر Lisharb (رینکه، ۲۰۱۱)، SHTns (شفر، ۲۰۱۳) و SHTOOLS (ویکزورک و مچد، ۲۰۱۸) به کار گرفته شده است. معادلات این روش عبارت‌اند از (فوکوشیما، ۲۰۱۲):

$$P_{n,m} = a_{nm} t P_{n-1,m} - b_{nm} P_{n-2,m}, \quad n \geq 2, \quad 0 \leq m \leq n-2$$

$$a_{nm} = \sqrt{\frac{(2n-1)(2n+1)}{(n-m)(n+m)}}, \quad b_{nm} = \sqrt{\frac{(2n+1)(n+m-1)(n-m-1)}{(2n-3)(n-m)(n+m)}} \quad (5)$$

مقدار اولیه برای شروع تکرار، $P_{m,m}$ است که از رابطه بازگشتی زیر محاسبه می‌شود:

$$P_{m,m} = \sqrt{\frac{2m+1}{2m}} u P_{m-1,m-1}, \quad P_{0,0} = 1, \quad n \geq 1 \quad (6)$$

که $u = \sin \theta$ است. در رابطه (۶) با نزدیک شدن به قطب، u کوچک می‌شود. با افزایش m مقادیر $P_{m,m}$ به طور متوالی کوچک می‌شوند به طوری که از بازه اعداد با دقت مضاعف قابل ذخیره در استاندارد IEEE754 برای حساب ممیز شناور خارج می‌شوند. در این استاندارد، اعداد با دقت مضاعف در بازه $[10^{-308}, 10^{+308}]$ قرار می‌گیرند. با خارج شدن از این بازه از سمت پایین و بالا به ترتیب خطای عددی پاریز و سرریز رخ می‌دهد. در این حالت مقادیر عددی $P_{m,m}$ (به اشتباه) صفر در نظر گرفته می‌شوند و رابطه (۵) به نتایج غلط می‌انجامد.

$$\begin{cases} C_{nm} \\ S_{nm} \end{cases} = \frac{1}{4\pi} \iint_S f(\theta, \lambda) P_{nm}(t) \begin{cases} \cos m\lambda \\ \sin m\lambda \end{cases} ds, \quad (8)$$

که $ds = \sin \theta d\theta d\lambda$ عنصر مساحت است. شکل گسسته رابطه فوق عبارت است از:

$$\begin{cases} C_{nm} \\ S_{nm} \end{cases} = \frac{1}{4\pi} \sum_{i=1}^I \sum_{j=1}^J \omega_{ij} f(\theta_i, \lambda_j) \begin{cases} Y^c(\theta_i, \lambda_j) \\ Y^s(\theta_i, \lambda_j) \end{cases}. \quad (9)$$

که ω_{ij} ضرایب انتگرال گیری و $f(\theta_i, \lambda_j)$ داده‌ها در موقعیت شبکه است. رابطه اخیر را می‌توان به صورت زیر بازنویسی کرد:

$$\begin{cases} C_{nm} \\ S_{nm} \end{cases} = \frac{1 + \delta_{m0}}{4} \sum_{i=1}^I w_i P_{nm}(t_i) \begin{cases} A_m(\theta_i) \\ B_m(\theta_i) \end{cases} \quad (10)$$

که $A_m(\theta_i)$ و $B_m(\theta_i)$ برابرند با:

$$\begin{cases} A_m(\theta_i) \\ B_m(\theta_i) \end{cases} = \frac{1}{(1 + \delta_{m0})\pi} \sum_{j=1}^J f(\theta_i, \lambda_j) \begin{cases} \cos m\lambda_j \\ \sin m\lambda_j \end{cases}. \quad (11)$$

استفاده از معادلات (۱۰) و (۱۱) امکان استفاده از تبدیل فوریه را برای محاسبه ضرایب $A_m(\theta_i)$ و $B_m(\theta_i)$ فراهم می‌کند که نقش مهمی در افزایش سرعت محاسبات دارد. روابط مورد نیاز در شکل ۱ آمده است.

$$d_{nm} = \frac{a_n}{2n} \sqrt{(n-m)(n-m-1)} \quad (7)$$

$$e_{nm} = \frac{a_n}{2n} \sqrt{1 + \delta_{m-1,0}} \sqrt{(n+m)(n+m-1)}$$

زینگ و همکاران (۲۰۲۰) نشان دادند دقت و سرعت محاسبات روش بلیکوف از روش ERA بیشتر است. علاوه بر کارایی بیشتر، روش بلیکوف فرمول‌بندی بسیار ساده‌تری دارد و بدون نیاز به الگوریتم اضافی، تا درجات بسیار بالا نظیر $N = ۶۴۸۰۰$ می‌تواند fnALF را محاسبه کند.

فرض کنید توابع P_{nm} برای درجه و مرتبه‌های مختلف به شکل ماتریس پایین‌مثلثی باشند به طوری که سطرها بیانگر درجه n و ستون‌ها بیانگر مرتبه m باشند. در این صورت روش‌های ERA و بلیکوف، توابع fnALF را به ترتیب به صورت سطری و ستونی محاسبه می‌کنند که با توجه به نحوه عمل متفاوت آنها، الگوریتم‌های SHA/SHS مختلفی خواهند داشت؛ لذا مقایسه عملکرد دو روش فوق، به تنهایی کافی نیست و باید کل فرایند SHA مقایسه شود.

۳ اجرای SHT

در تحلیل هماهنگ، ضرایب C_{nm} و S_{nm} از رابطه زیر به دست می‌آیند:

<pre> for i = 1 to n_φ ! n_φ= number of φ F_i = FFT{f(φ_i, λ_{1:2N})} ! f(φ_i, λ_{1:2N}): data of parallel φ_i A_m = Re (F_i) . w_i !w_i : integrations weights B_m = -Im(F_i) . w_i N_φ = N cos φ + μ' ! polar optimization for m = 0 to N_φ P_{m:N,m} = call ALF_col(φ, m) for n = m to N C_{nm} = C_{nm} + P_{nm} A_n S_{nm} = S_{nm} + P_{nm} B_n end !for n end !for m end !for i </pre>	<pre> for i = 1 to n_φ ! n_φ= number of φ F_i = FFT{f(φ_i, λ_{1:2N})} ! f(φ_i, λ_{1:2N}): data of parallel φ_i A_m = Re (F_i) . w_i !w_i : integrations weights B_m = -Im(F_i) . w_i for n = 0 to N m_φ = n cos φ + μ ! polar optimization P_{n,0:m_φ} = call ALF_row(φ, n) for m = 0 to m_φ C_{nm} = C_{nm} + P_{nm} A_n S_{nm} = S_{nm} + P_{nm} B_n end !for m end !for n end !for i </pre>
---	--

شکل ۱. الگوریتم تحلیل کروی به کاررفته در این مطالعه برای محاسبات ستونی fnALF (سمت راست) و محاسبات سطری (سمت چپ).

مشخصات شبکه $f(\theta_i, \lambda_j)$ (موقعیت نقاط و تعداد آنها در راستای طول و عرض) به نظریه نمونه برداری روی کره وابسته است. در این مقاله از روش گوس- لژاندر برای SHA استفاده می‌شود. برای تحلیل هماهنگ به روش گوس- لژاندر داده‌ها باید روی شبکه گوس معلوم باشند. موقعیت نقاط این شبکه در راستای عرض، ریشه‌های توابع لژاندر است (اسنیو، ۱۹۹۴):

ادامه می‌یابد (بالمینو و همکاران، ۲۰۱۲). همانند محاسبات تحلیل، از تبدیل فوریه برای سرعت بخشیدن به محاسبات SHS نیز می‌توان استفاده کرد. برای این منظور رابطه (۱) به صورت زیر بازنویسی می‌شود:

$$f(\theta_i, \lambda_j) = \sum_{m=0}^N \sum_{n=m}^N C_{nm} Y_{nm}^c(\theta_i, \lambda_j) + S_{nm} Y_{nm}^s(\theta_i, \lambda_j) \quad (16)$$

این رابطه را می‌توان به دو رابطه زیر تبدیل کرد:

$$f(\theta_i, \lambda_j) = \sum_{m=0}^N C_m(\theta_i) \cos m\lambda_j + D_m(\theta_i) \sin m\lambda_j \quad (17)$$

$$\begin{Bmatrix} C_m(\theta_i) \\ D_m(\theta_i) \end{Bmatrix} = \sum_{n=m}^N P_{nm}(\theta_i) \begin{Bmatrix} C_{nm} \\ S_{nm} \end{Bmatrix}.$$

در محاسبات SHS ابتدا ضرایب C_m و D_m برای هر مدار θ_i محاسبه می‌شود سپس از رابطه (۱۷) مقدار $f(\theta_i, \lambda_j)$ با تبدیل معکوس فوریه به دست می‌آید. شکل ۲ الگوریتم محاسباتی SHS را برای محاسبه ستونی و سطری fnALF ارائه می‌کند.

$$P_N(t_i) = 0, \quad i = 1, 2, 3, \dots, N + 1. \quad (12)$$

وزن انتگرال گیری برابر است با:

$$w_i = \frac{2}{[t_i P'_n(\cos \theta_i)]^2} \quad (13)$$

که P' مشتق تابع لژاندر نسبت به t است. توزیع نقاط روی طول جغرافیایی نیز به صورت زیر است:

$$\lambda_j = j\Delta, \quad j = 0, 1, \dots, 2N \quad \Delta = \frac{\pi}{N} \quad (14)$$

تحلیل هماهنگ روی شبکه گوس توانایی بازسازی کامل مشاهدات را دارد. شکل ۱ الگوریتم تحلیل کروی را برای روش‌های ستونی و سطری محاسبه fnALF نشان می‌دهد. نکته گفتنی در این الگوریتم‌ها، بهینه‌سازی الگوریتم برای مناطق قطبی است. جکلی و همکاران (۲۰۰۷) نشان دادند:

$$P_{nm} \rightarrow 0, \quad m > n \sin \theta + \mu \quad (15)$$

که μ پارامتری تجربی است. در این مطالعه مقدار $N = 10 \times 10^3$ با $\mu = 10 \times 10^3$ با سعی و خطا انتخاب شده است. با استفاده از رابطه (۱۵) در روش سطری محاسبه fnALF، محاسبات در هر سطر به جای مرتبه n تا مرتبه

<pre> for i = 1 to n_φ N_φ = N cos φ + μ ! polar optimization for m = 0 to N_φ P_{m:N,m} = call ALF(φ, m) for n = m to N C_m = C_m + C_{nm} P_{nm} D_m = D_m + S_{nm} P_{nm} End !for n end !for m f(i, 1:n_λ) = Re{FFT⁻¹(C_m + iD_m)} end !for i </pre>	<pre> for i = 1 to n_φ for n = 0 to N m_φ = N cos φ + μ' ! polar optimization P_{n,0:m_φ} = call ALF(φ, n) for m = 0 to m_φ C_m = C_m + C_{nm} P_{nm} D_m = D_m + S_{nm} P_{nm} end !for m end !for n f(i, 1:n_λ) = Re{FFT⁻¹(C_m + iD_m)} end !for i </pre>
---	--

شکل ۲. الگوریتم تحلیل کروی به کاررفته در این مطالعه برای محاسبات ستونی fnALF (سمت راست) و محاسبات سطری (سمت چپ).

۴ محاسبات عددی

برای محاسبات عددی این مطالعه یک نرم‌افزار با زبان برنامه‌نویسی فرترن ۹۵ توسعه داده شد. برای روش ERA از نرم‌افزار توسعه‌داده‌شده فوکوشیما (۲۰۱۲) استفاده شد. روش‌های بلیکوف و محاسبات SHS و SHA را نویسنده توسعه داده است. از آنجاکه اغلب پردازنده‌های امروزی قادر به رایانش موازی به صورت چندنخی هستند، در نرم‌افزار توسعه‌داده‌شده، به کمک توابع کتابخانه‌ای Openmp، امکان رایانش موازی وجود دارد. محاسبات SHS و SHA شامل سه حلقه تکراری تودرتو است که برای دستیابی به کارایی بهینه، محاسبات روی حلقه اول یا حلقه دوم می‌توانند به‌طور موازی انجام شوند (شکل‌های ۱ و ۲). در اغلب نرم‌افزارهای موجود تحلیل هماهنگی که از محاسبات ستونی fnALF بهره می‌برند، حلقه دوم به‌طور موازی محاسبه می‌شود درحالی‌که در روش سطری (روش بلیکوف) امکان رایانش موازی روی حلقه دوم منتفی است؛ زیرا محاسبه هر سطر وابسته به سطر قبلی است. این یک نکته منفی در به‌کارگیری روش بلیکوف در SHT است؛ چون اعمال موازی‌سازی حلقه دوم به‌مراتب از حلقه اول ساده‌تر است. در نرم‌افزار توسعه‌داده‌شده در این مطالعه، محاسبات حلقه اول (حلقه روی عرض کروی) به‌طور موازی انجام می‌شود. در این نرم‌افزار برای افزایش سرعت از ویژگی تقارن fnALF در نیمکره شمالی نسبت به نیمکره جنوبی نیز استفاده شده است؛ بنابراین توابع fnALF تنها برای یک نیمکره محاسبه می‌شوند که این موضوع تأثیر چشمگیری در افزایش سرعت محاسبات دارد. برای جزئیات بیشتر به ویکتورک و مچد (۲۰۱۸) مراجعه کنید.

بخش اول مقایسه دو روش سطری و ستونی، به کارایی آنها در محاسبه fnALF اختصاص دارد. منظور از کارایی، دو عامل سرعت و دقت محاسبات است. همان‌طور که در قبل اشاره شد، با نزدیک شدن به قطب، محاسبه fnALF

مشکلات عددی همراه است؛ لذا نقاط آزمون باید طوری انتخاب شوند که مشکلات عددی احتمالی روش‌ها در آنها نمایان شود. برای این منظور، محاسبات fnALF روی ۸۹۹ عرض جغرافیایی انجام گرفت که به ترتیب زیر تعریف می‌شوند:

$$\varphi = \frac{\Delta}{2} + (i - 1) \times 0.1, \quad (18)$$

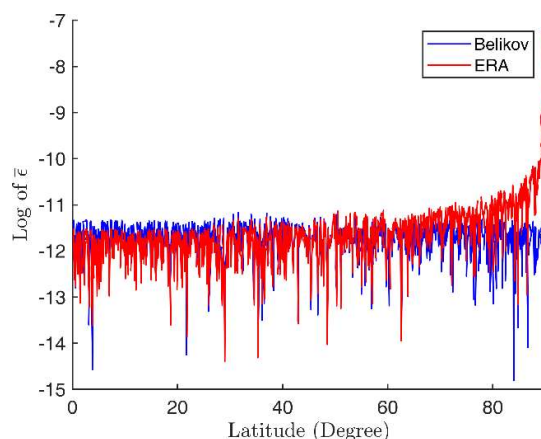
$$i = 1, 2, \dots, 899$$

در این رابطه $\Delta = \frac{N}{180^\circ}$ است. علت جابه‌جایی شبکه (۱۸) به اندازه $\frac{\Delta}{2}$ این است که در SHT تا درجه/مرتبه N، به محاسبه $P_{nm} \left(\cos \left(\frac{\pi}{2} - \frac{\Delta}{2} \right) \right)$ نیاز است. برای بررسی دقت (نسبی) fnALF از اتحاد زیر استفاده می‌شود (فوکوشیما، ۲۰۱۲):

$$\delta I = \left| \frac{I}{(N+1)^2} - 1 \right|; \quad I = \sum_{n=0}^N \sum_{m=0}^n [P_{nm}(\cos \theta)]^2 = (N+1)^2 \quad (19)$$

دقت هر دو روش ERA و بلیکوف در مطالعه زینگ و همکاران (۲۰۲۰) بررسی و اثبات شده است. علت استفاده از رابطه (۱۹)، آزمون صحت نرم‌افزارهای توسعه‌داده‌شده است. شکل ۳ تابع δI را برای درجه $N=64800$ نشان می‌دهد. بر اساس این شکل، دقت روش ERA برخلاف روش بلیکوف با نزدیک شدن به قطب کاهش چشمگیری دارد. البته برای عرض‌های کوتاه $40^\circ < \varphi$ دقت روش ERA کمی بیشتر از بلیکوف است. این نتایج با نتایج مطالعات زینگ و همکاران (۲۰۲۰) همخوانی دارد.

جدول ۱ زمان محاسبات را برای محاسبه شبکه رابطه (۱۸) به دو روش ERA و بلیکوف در دو حالت سری و موازی برای درجات ۲۱۶۰ تا ۶۴۸۰۰ ارائه می‌دهد. دو علت شروع از درجه ۲۱۶۰ این است که مشکلات عددی در محاسبه fnALF اغلب از درجه ۲۱۶۰ شروع می‌شوند و دیگر اینکه معمولاً محاسبات برای درجات پایین سرعت زیادی دارند. محاسبات عددی این مطالعه در یک رایانه



شکل ۳. مقدار کمیت. δI در عرض‌های مختلف برای دو روش بلیکوف و ERA در درجه هماهنگ $N=64800$.

جدول ۱. زمان محاسبات finALF (برحسب ثانیه) برای درجات مختلف در رایانش سری و موازی.

Nmax	۲۱۶۰	۵۴۰۰	۱۰۸۰۰	۲۱۶۰۰	۴۳۲۰۰	۶۴۸۰۰
Nthread=1						
بلیکوف	۲/۸۸۸	۱۴/۶۷۸	۶۳/۳۹۵	۲۶۱/۲۱	۱۱۲۵/۸۶	۲۵۰۸/۹۲
ERA	۵/۷۱۵	۳۳/۲۰۱	۱۳۲/۵۹	۵۳۴/۷۵	۲۱۹۹/۴۸	۴۸۵۳/۹۹
Nthread=10						
بلیکوف	۰/۵۷۴	۳/۹۲۸	۱۹/۵۶۵	۹۴/۷۶۳	۴۲۸/۸۵	۱۰۱۰/۴۱
ERA	۰/۷۹۵	۴۰/۸۶۰	۱۹/۸۱۹	۸۰/۲۷۱	۳۲۵/۷۰	۷۳۲/۴۰

نکته درخور توجه این است که برای درجات بالاتر از 10800 زمان محاسبات روش ERA کمتر از روش بلیکوف است.

همان‌طور که در شکل‌های ۱ و ۲ مشهود است، برای افزایش سرعت در محاسبات SHT، از بهینه‌سازی در مناطق قطبی استفاده می‌شود. برای بررسی اثر بهینه‌سازی بر کارایی روش‌ها، زمان محاسبات با بهینه‌سازی در جدول ۲ ارائه شده است. نتایج عددی جدول ۲ نشان می‌دهد بهینه‌سازی قطبی روی روش‌های سطری کارایی بیشتری نسبت به روش‌های ستونی دارد. همچنین بهینه‌سازی، ضعف روش بلیکوف را در رایانش موازی برطرف می‌کند به طوری که سرعت محاسبات در روش بلیکوف تقریباً $1/5$ برابر روش ERA در تمامی درجات 2160 تا 64800 است.

شخصی با پردازنده Core-i7 8700K و حافظه 40GB انجام شده است. کامپایلر مورد استفاده gfortran v7.4 تحت فرمان `march=native -O3` اجرا شده است. در رایانش موازی از ده نخ پردازشی (از مجموع دوازده نخ پردازشی موجود) استفاده شده است. نتایج عددی جدول ۱ نشان می‌دهد روش بلیکوف در حالت سری برای درجات 2160 و بالاتر بیش از دو برابر سریع‌تر از روش ERA است. این نسبت برای همه درجات 2160 تا 64800 تقریباً یکسان است. در حالت موازی، نتایج به درجه N وابسته است. میزان افزایش سرعت محاسبات با ده نخ پردازشی نسبت به یک نخ پردازشی در روش‌های بلیکوف و ERA به ترتیب حدود $7/6$ و $3/3$ برابر است. می‌توان گفت روش بلیکوف نسبت به ERA عملکرد ضعیفی در محاسبات موازی دارد.

جدول ۲. زمان محاسبات fnALF (برحسب ثانیه) برای درجات مختلف با اعمال بهینه‌سازی قطبی در رایانش سری و موازی.

Nmax	۲۱۶۰	۵۴۰۰	۱۰۸۰۰	۲۱۶۰۰	۴۳۲۰۰	۶۴۸۰۰
Nthread=۱						
بلیکوف	۱/۵۵۷	۹/۳۶۲	۳۹/۲۵۰	۱۶۶/۵۳	۷۱۸/۵۳	۱۶۳۸/۶۴
ERA	۳/۵۷۲	۲۲/۲۳۵	۹۰/۴۲۴	۳۶۶/۹۶	۱۴۸۶/۹۶	۳۳۶۶/۱۶
Nthread=۱۰						
بلیکوف	۰/۲۵۵	۱/۵۶۰	۶/۶۲۵	۳۱/۰۳۳	۱۴۱/۴۰۸	۳۳۳/۷۰۸
ERA	۰/۴۴۸	۲/۹۸۰	۱۲/۴۳۲	۵۱/۰۴۹	۲۰۹/۹۶۶	۴۷۵/۷۲۱

جدول ۳. خطای جذر میانگین مربعات و خطای بیشینه SHT برای درجات مختلف.

Nmax	۲۱۶۰	۵۴۰۰	۱۰۸۰۰	۲۱۶۰۰
e_{max}				
بلیکوف	۰/۶۷E-۱۱	۰/۲۱E-۱۰	۰/۳۸E-۱۰	۰/۸۲E-۱۰
ERA	۰/۲۳E-۱۱	۰/۷۸E-۱۱	۰/۳۸E-۱۰	۰/۱۱E-۰۹
RMS				
بلیکوف	۰/۴۶E-۱۲	۰/۱۲E-۱۱	۰/۸۷E-۱۲	۰/۱۶E-۱۱
ERA	۰/۱۵E-۱۲	۰/۳۹E-۱۲	۰/۸۶E-۱۲	۰/۱۶E-۱۱

نتایج عددی دقت محاسبات در جدول ۳ آمده است. با توجه به مشکلات حافظه، محاسبات تا درجه/مرتب ۲۱۶۰۰ متوقف شده است. درجه هماهنگ ۴۳۲۰۰ در یارانش موازی به بیش از ۷۵ GB حافظه RAM نیاز دارد که در دسترس نبود. با توجه به سهم اندک مناطق قطبی، علی‌رغم اینکه دقت روش سطری بلیکوف در مناطق قطبی از ERA بیشتر است، درعمل خطای جذر میانگین مربعات برای درجات ۱۰۸۰۰ و ۲۱۶۰۰ در هر دو روش یکسان است (جدول ۳). به عبارت دیگر، دقت زیاد روش بلیکوف در محاسبه fnALF در عرض‌های متوسط و قطبی، به افزایش دقت SHT منجر نمی‌شود. از نظر خطای جذر میانگین مربعات، هر دو روش تا درجه ۲۱۶۰۰ دقت یکسانی دارند. البته افزایش خطای بیشینه در درجه ۲۱۶۰۰ در روش ERA مشهود است.

با توجه به یکسان بودن نتایج دقت هر دو روش، زمان محاسبات، پارامتری مهم در محاسبات SHT است. زمان محاسبات SHS و SHA در حالت رایانش سری و موازی

همان‌طور که پیش از این اشاره شد، بهتر است کارایی دو روش سطری بلیکوف و ستونی ERA در محاسبه fnALF، در محاسبات SHA و SHS بررسی شود. برای این منظور، در این مطالعه از روشی استفاده می‌شود که در مطالعه شفر (۲۰۱۳) ارائه شده است. ابتدا ضرایب C_{nm} و S_{nm} برای درجات مختلف به صورت تصادفی در بازه $[-1, +1]$ تولید می‌شوند. مقدار تابع روی یک شبکه گوس با SHS تولید می‌شود و ضرایب هماهنگ C'_{nm} و S'_{nm} با SHA به روش گوس-لژاندر بازیابی می‌شوند. نرم L_2 (خطای جذر میانگین مربعات) و نرم چیشف (خطای بیشینه) معیارهای بررسی دقت این فرایند هستند:

$$RMS^2 = \frac{2}{(N+1)(N+2)} \sum_{n=0}^N \sum_{m=0}^N (c_{nm} - c'_{nm})^2 + (s_{nm} - s'_{nm})^2, \quad (20)$$

$$e_{max} = \max(|c_{nm} - c'_{nm}|, |s_{nm} - s'_{nm}|)$$

این فرایند بسته، یک بار با الگوریتم سطری و یک بار با الگوریتم ستونی برای درجات مختلف انجام می‌شود.

الگوریتم با بیشترین کارایی انتخاب شده است. در این راستا سعی شده است تا حد امکان از توانایی محاسبات برداری در FORTRAN95 استفاده شود که نقش چشمگیری در افزایش سرعت محاسبات دارد. سرعت اجرای برنامه به نحوه ذخیره‌سازی ضرایب S_{nm} و C_{nm} نیز بستگی دارد. برای این منظور می‌توان از آرایه‌های دوبعدی (به شکل یک ماتریس) و آرایه‌های یک‌بعدی (به شکل یک بردار) استفاده کرد. بر اساس محاسبات این مطالعه، استفاده از آرایه‌های دوبعدی در محاسبات سطری SHT سرعت محاسبات را تا ده برابر کاهش می‌دهد، درحالی‌که تأثیر درخور توجهی بر زمان محاسبات ستونی ندارد. علاوه بر این، ضرایب S_{nm} و C_{nm} در حالت دوبعدی شبیه ماتریس‌های پایین‌مثنی هستند. استفاده از آرایه‌های دوبعدی به فضای حافظه‌ای دو برابر حالت آرایه یک-بعدی نیاز دارد؛ لذا نرم‌افزار توسعه داده شده در این مطالعه از آرایه‌های یک‌بعدی در هر دو الگوریتم سطری و ستونی استفاده می‌کند.

با ده نخ پردازشی بر اساس الگوریتم برای درجه‌های مختلف در جدول ۴ آمده است. مطابق نتایج این جدول، در حالت سری، سرعت SHS و SHA با الگوریتم سطری به ترتیب $1/8$ و $1/7$ برابر الگوریتم ستونی است، اما در رایانش موازی، سرعت محاسبات SHS الگوریتم سطری حدود 10% بیشتر از الگوریتم ستونی است. همچنین سرعت محاسبات SHA الگوریتم سطری به طور متوسط 20% بیشتر از الگوریتم ستونی است. این موضوع بیانگر آن است که روش بلیکوف در محاسبات موازی همانند محاسبات سری باعث افزایش نزدیک به دو برابری محاسبات SHT نمی‌شود.

زمان محاسبات SHT به الگوریتم و زبان برنامه‌نویسی وابسته است. در توسعه نرم‌افزار این مطالعه از زبان فرترن استفاده شده است که در کنار زبان برنامه‌نویسی C/C++ از سریع‌ترین زبان‌های برنامه‌نویسی در محاسبات عددی است. همچنین در نرم‌افزار توسعه داده شده، الگوریتم‌های مختلفی برای محاسبات SHT مقایسه و در نهایت،

جدول ۴. زمان محاسبات SHT (برحسب ثانیه) برای درجات مختلف (صورت کسر، زمان SHS و منخرج کسر، زمان SHA است).

Nmax	۲۱۶۰	۵۴۰۰	۱۰۸۰۰	۲۱۶۰۰
Nthread=۱				
بلیکوف	۴/۶	۶۳/۵	۴۹۸/۹	۳۹۴۶/۴
	۵/۵	۷۶/۵	۶۱۸/۴	۵۰۳۹/۵
ERA	۷/۸	۱۱۷/۶	۹۰۸/۱	۷۱۷۸/۸
	۸/۶	۱۳۲/۹	۱۰۸۲/۶	۸۶۹۰/۳
Nthread=۱۰				
بلیکوف	۱/۳	۲۰/۰	۱۶۳/۲	۱۳۳۸/۰
	۲/۸	۴۰/۶	۳۵۶/۰	۲۷۳۲/۸
ERA	۱/۴	۲۲/۰	۱۷۶/۳	۱۴۸۷/۱
	۳/۳	۵۰/۴	۴۴۶/۶	۳۳۷۷/۰

۵ نتیجه‌گیری

الگوریتم سطری SHT بر مبنای محاسبه fnALF به روش بلیکوف است. در این مطالعه یک نرم‌افزار با توان رایانش موازی برای انجام دادن محاسبات SHS و SHA به روش GL در حالت ستونی و سطری توسعه داده شد. نتایج

در این مطالعه کارایی الگوریتم محاسبات ستونی و سطری SHT در رایانش سری و موازی مقایسه شد. الگوریتم ستونی SHT بر مبنای محاسبه fnALF به روش ERA و

روش سطری SHS و SHA با توجه به سرعت و دقت بیشتر نسبت به سایر روش‌ها توصیه می‌شود.

منابع

- Balmino, G., Vales, N., Bonvalot, S., and Briais, A., 2012, Spherical harmonic modelling to ultra-high degree of Bouguer and isostatic anomalies: *Journal of Geodesy*, **86**(7), 499–520, <https://doi.org/10.1007/s00190-011-0533-0534>.
- Belikov, M. V., 1991, Spherical harmonic analysis and synthesis with the use of column-wise recurrence relations: *Manuscripta Geodaetica*, **16**(6), 384–410.
- Bucha, B., and Janák, J., 2013, A MATLAB-based graphical user interface program for computing functionals of the geopotential up to ultra-high degrees and orders: *Computers & Geosciences*, **56**, 186–196, <https://doi.org/https://doi.org/10.1016/j.cageo.2013.03.012>.
- Enomoto, T., 2015, Comparison of Computational Methods of Associated Legendre Functions: *SOLA* **11**, 144–149.
- Fukushima, T., 2012, Numerical computation of spherical harmonics of arbitrary degree and order by extending exponent of floating point numbers: *Journal of Geodesy*, **86**(4), 271–285.
- Heiskanen, W. H., and Moritz, H., 1967, *Physical Geodesy*: W.H. Freeman and Co., San Francisco.
- Holmes, S. A., and Featherstone, W., 2002, A unified approach to the Clenshaw summation and the recursive computation of very high degree and order normalised associated Legendre functions: *Journal*

عددی نشان داد که سرعت روش بلیکوف در محاسبه fnALF در رایانش سری و موازی نزدیک به دو برابر روش ERA در درجات ۲۱۶۰ تا ۶۴۸۰۰ است. برای بررسی تأثیر کارایی روش محاسبه fnALF بر محاسبات SHT، محاسبات با استفاده از ضرایب شبیه‌سازی شده در یک چرخه بسته برای درجات مختلف از ۲۱۶۰ تا ۶۴۸۰۰ انجام شد. نتایج عددی نشان داد در پردازش سری برای SHS و SHA، سرعت محاسبات روش سطری حدود ۱/۷ برابر روش ستونی است. این در حالی است که در رایانش موازی با ده نخ پردازشی، بیشینه افزایش سرعت برای SHS و SHA در روش سطری نسبت به ستونی برابر ۲۵٪ است. بر اساس مقایسه RMS و از نظر دقت محاسبات، هر دو روش تقریباً دقت یکسانی دارند. البته بیشترین خطا در درجات بالایی نظیر ۲۱۶۰۰، در روش ERA از بلیکوف بیشتر است.

یک نکته در به کارگیری روش بلیکوف برای محاسبه fnALF استقلال نداشتن سطرها است. به عبارت دیگر، محاسبه هر سطر از P_{nm} وابسته به $P_{n-1,m}$ است. در این حالت محاسبات نمی‌تواند روی درجه n به‌طور موازی محاسبه شود. تنها امکان محاسبات به‌طور موازی روی پارامتر عرض کروی وجود دارد. نرم‌افزارهای فوق سریع SHT نظیر Libsharp و SHTns برای افزایش سرعت محاسبات علاوه بر موازی‌سازی، از توانایی یک دستور چندداده (SIMD) نیز سود می‌برند. استفاده از SIMD سرعت محاسبات را بسته به نوع CPU تا چندین برابر افزایش می‌دهد. در اغلب این نرم‌افزارها حلقه اول محاسبات SHT در شکل‌های ۱ و ۲ با SIMD و حلقه دوم به‌طور موازی محاسبه می‌شود که این امکان برای روش بلیکوف وجود ندارد. البته به کارگیری توانایی SIMD در محاسبات SHT بسیار پیچیده است و هیچ‌یک از نرم‌افزارهای ژئودتیکی از آن استفاده نمی‌کنند؛ بنابراین

- of Geodesy, **76**(5), 279-299.
- Jekeli, C., 2017, Spectral Methods in Geodesy and Geophysics: CRC Press.
- Jekeli, C., Lee, J. K., and Kwon, J. H., 2007, On the computation and approximation of ultra-high-degree spherical harmonic series: Journal of Geodesy, **8**(9), 603-615.
- Nesvadba, O., 2011, Nothing to Fear from Ellipsoidal Harmonics: General Assembly of the European Geosciences Union, Vienna, 3-8.
- Novikova, E., and Dmitrenko, A., 2016, Problems and methods of calculating the Legendre functions of arbitrary degree and order: Geodesy and Cartography, **65**(2), 283-132.
- Prezeau, G., and Reinecke, M., 2010, Algorithm for the evaluation of reduced Wigner matrices: The Astrophysical Journal Supplement Series, **190**(2), 267-274.
- Reinecke, M., 2011, LibpSHT-algorithms for efficient spherical harmonic transforms: Astronomy & Astrophysics, **526**, A108.
- Rexer, M., and Hirt, C., 2015, Ultra-high-degree surface spherical harmonic analysis using the Gauss-Legendre and the Driscoll/Healy quadrature theorem and application to planetary topography models of Earth, Mars and Moon: Surveys in Geophysics, **36**(6), 803-830, <https://doi.org/10.1007/s10712-015-9345-z>.
- Schaeffer, N., 2013, Efficient spherical harmonic transforms aimed at pseudospectral numerical simulations: Geochemistry, Geophysics, Geosystems, **14**(3), 751-58, <https://doi.org/10.1002/ggge.20071>.
- Sneeuw, N., 1994, Global spherical harmonic analysis by least-squares and numerical quadrature methods in historical perspective: Geophysical Journal International, **118**(3), 707-716, <https://doi.org/10.1111/j.1365-246X.1994.tb03995.x>.
- Tozer, B., Sandwell, D. T., Smith, W. H. F., Olson, C., Beale, J. R., and Wessel, P., 2019, Global bathymetry and topography at 15 arc Sec: SRTM15+: Earth and Space Science, **6**(10), 1847-1864.
- Wieczorek, M. A., and Meschede, M., 2018, SHTools: Tools for working with spherical harmonics: Geochemistry, Geophysics, Geosystems, **19**(8), 2574-2592, <https://doi.org/10.1029/2018gc007529>.
- Wittwer, T., Klees, R., Seitz, K., and Heck, B., 2008, Ultra-high degree spherical harmonic analysis and synthesis using Extended-Range Arithmetic: Journal of Geodesy, **82**(4), 223-229.
- Xing, Z., Li, S., Tian, M., Fan, D., and Zhang C. J., 2020, Numerical experiments on column-wise recurrence formula to compute fully normalized associated Legendre functions of ultra-high degree and order: Journal of Geodesy, **94**(1), id. 2.

Comparison of row-wise and column-wise methods in ultra-high degree/order spherical harmonic transforms

Mehdi Goli^{1*}

¹ Assistant Professor, Faculty of Civil Engineering, Shahrood University of Technology, Shahrood, Iran

(Received: 30 July 2020, Accepted: 22 December 2020)

Summary

Spherical harmonic transform (SHT) provides a standard tool for spectral analysis of data measured on a spherical manifold particularly in geodesy and geophysics, such as Earth's gravity/magnetic field, topography, atmosphere, etc. Thanks to recent satellite missions such as SRTM, the spatial resolution of geoscience data decreases, leading to an increase in the degree/order of spherical harmonic. The accurate and fast computation of fully normalized associated Legendre functions (fnALF) is the main part of spherical harmonic computations. In polar region, for high degrees ($n > 2000$), the arithmetic underflow may occur and Extended Range Arithmetic (ERA) can be applied to fix underflow problem. In Belikov method the four terms row-wise recurrence formula is used for fnALF computation without any numerical issues.

Previous studies have only addressed the efficiency of the ERA and Belikov method without mentioning to their performance in spherical harmonic synthesis (SHS) and analysis (SHA). This study aims at investigating the effect of row-wise and column-wise fnALF on SHS/SHA for high harmonic degree/order. For this purpose, we developed parallel FORTRAN95 software that utilizes Belikov and ERA to perform fnALF in the row-wise and column-wise SHT, respectively. For SHA, exact Gauss-Legendre quadrature was used that allows one to determine the level error of SHT.

In the first part of numerical results, the computation time and error of Belikov and ERA methods for calculating fnALF are compared in serial and parallel processing. Based on the results of the developed software, Belikov method is approximately 2 times faster than ERA for degrees 2160 to 64800 in both serial and parallel mode. Also for mid-latitudes, the errors for both methods are nearly the same. In polar region the error in ERA grows, while Belikov method has not latitudinal dependencies. To evaluate the performance of fnALF method on error in SHS/SHA, calculations were performed using random spherical harmonic coefficients (C_{nm} , S_{nm}) between $[-1, +1]$. Using SHS and SHA in a closed cycle, the computed coefficients (C'_{nm} , S'_{nm}) were compared with original coefficients (C_{nm} , S_{nm}). The RMS and maximum error for different degrees from $N=2160$ to 21600 were computed. The level error of two methods in terms of RMS is in the same level. Numerical results show that in serial mode, the row-wise SHS/SHA is approximately 1.8 times faster than column-wise method. However, in parallel computing with 10 threads, the speed up factor decreases to about 1.2.

Keywords: Associated Legendre function, spherical harmonic transforms, Belikov method, Extended Range Arithmetic (ERA)

*Corresponding author:

goli@shahroodut.ac.ir