JOURNAL OF COMPUTATIONAL PHYSICS 80, 506–511 (1989)

Note

A Test of a Modified Algorithm for Computing Spherical Harmonic Coefficients Using an FFT

1. Introduction

This note describes the results of testing a modified method based on an algorithm by Dilts [1] for computing the spherical harmonic expansion coefficients for a function on a sphere using a two-dimensional FFT. The new version of the Dilts program eliminates problems of overflow and large storage requirements encountered when harmonic degree values of $l > 16$ are computed, thus allowing the determination of coefficients for high l values ($l \leqslant 300$). However, results from timing tests indicate that the new version of the Dilts program is not practical for computing spherical harmonic expansion coefficients for large l due to the very long CPU times required for execution of the program. The most effective technique remains the one pioneered by Brown [2].

The development of helioseismology [3], wherein the oscillations of the sun are analyzed to infer the physical state of the solar interior, has created a need for a fast spherical harmonic transformation procedure that can be used at high values of l, the spherical harmonic degree. In the context of the GONG (Global Oscillation Network Group) project, observations of the full solar disk with a resolution of 256×256 pixels will be obtained every minute nearly continuously for three years. Such observations can in principle provide information on global modes of oscillations with $l \leqslant 300$. Since for each mode with a given l there are $2l + 1$ nondegenerate modes with differing values of m, the azimuthal order, the projection of the data onto some 1.8×10^5 spherical harmonics must be performed each minute to avoid a huge data backlog. This is in addition to other processing that must be done on each image. A proposed experiment on the SOHO spacecraft would produce an even larger amount of data, obtaining images with 1024×1024 pixel resolution. There is thus great interest in the solar physics community in the development of fast spherical harmonic transforms. Recently, an algorithm appeared in the literature that held the potential of satisfying these stringent requirements [1]. Here we report on a modification to the algorithm that permitted its use at high l values, and on its performance.

The object of a spherical harmonic transform is to determine the coefficients F_{lm} of the expansion of a data image $f(\theta, \phi)$ on a sphere,

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} F_{lm} Y_{lm}(\theta, \phi), \tag{1}$$

0021-9991/89 $3.00

where F_{lm} is given by

$$F_{lm} = \int_{-\pi}^{\pi} \int_{0}^{\pi} Y_{lm}^{*}(\theta, \phi) f(\theta, \phi) \sin\theta \, d\theta \, d\phi. \tag{2}$$

The Dilts algorithm is based on a Fourier representation of the spherical harmonic functions,

$$Y_{lm}(\theta, \phi) = C_{lm} P_l^m(\cos\theta) \, e^{im\phi} \tag{3}$$

in which $P_l^m(\cos\theta)$ is the associated Legendre function of harmonic degree l and order m. The normalization coefficient C_{lm} is given by

$$C_{lm} = \left[\frac{(2l+1)(l-m)!}{4\pi(l+m)!} \right]^{1/2}. \tag{4}$$

In the Dilts method, Y_{lm} is represented by a two-dimensional Fourier series,

$$Y_{lm}(\theta, \phi) = \sum_{k=-l}^{l} B_k^{lm} e^{i(k\theta + m\phi)} \tag{5}$$

in which $e^{i(k\theta + m\phi)}$ is a single Fourier component and B_k^{lm} is the Fourier expansion coefficient.

The Dilts method for computing the B_k^{lm} involves a complex quantity a_k^{lm}, the coefficient resulting from the Fourier expansion of the P_l^m. When $l \neq m$, these a_k^{lm} are computed using the recursion relation

$$a_k^{l+1,m} = \tfrac{1}{2}[a_{k-1}^{lm} + a_{k+1}^{lm} + i(l+m)(a_{k-1}^{l,m-1} - a_{k+1}^{l,m-1})] \tag{6}$$

when $k \neq l$, and the boundary relation

$$a_l^{l+1,m} = \tfrac{1}{2}[a_{l-1}^{lm} + i(l+m)\, a_{l-1}^{l,m-1}] \tag{7}$$

when $k = l$. When $l = m$, the relation

$$a_{l-2j}^{ll} = \frac{i^l (2l)!}{4^l (l-j)!\, j!} \tag{8}$$

is used. Seed values for values of l, m, and k of 0 and 1 are used to initialize the recursion process, and the B_k^{lm} are finally computed using

$$B_k^{lm} = C_{lm} a_k^{lm}. \tag{9}$$

After the B_k^{lm} are computed, the image data f is interpolated from the observed $(\sin\theta, \sin\phi)$ grid onto an evenly spaced grid in θ and ϕ and is then transformed via a two-dimensional FFT,

$$f(\theta, \phi) = \sum_{a=-N}^{N} \sum_{b=-N}^{N} f_{ab} e^{i(a\theta + b\phi)}, \tag{10}$$

where N is the number of pixels along the two dimensions of the image. Here, the image is square, but this is not required. The desired spherical harmonic coefficients F_{lm} are then determined from

$$F_{lm} = -\pi \sum_{k=-l}^{l} B_k^{lm*} \left[2 \sum_{a=-N}^{k-2} \frac{[(-1)^{a-k}+1]}{(a-k)^2-1} f_{am} + i\pi f_{k-1,m} - 4f_{k,m} \right.$$
$$\left. - i\pi f_{k+1,m} + 2 \sum_{a=k+2}^{N} \frac{[(-1)^{a-k}+1]}{(a-k)^2-1} f_{am} \right] \tag{11}$$

Dilts calculated the B_k^{lm} once and then stored them, using them in Eq. (11) as needed.

2. Program Modifications

We implemented the Dilts algorithm on a VAX 8600 computer. Two problems immediately arose when attempting to use the algorithm for high l values ($l > 16$). The first problem was the large storage space required to keep the entire set of B_k^{lm}. The second problem was the overflows generated by the factorials in C_{lm} and a_{l-2j}^{ll}.

The amount of storage space required for the program should be substantially less than the memory available in the core of the computer, since the i/o time required for page faulting on a virtual memory computer such as a VAX is much greater than the average floating point operation execution time. For $l = 250$, the amount of storage space required is about 5.3 Mb. This space was greatly reduced by noticing that the recursion relations require storing only the coefficients for the previous l value. Thus, by keeping only the coefficients for the previous l value and recalculating the B_k^{lm} as needed rather than storing the entire set, the storage space was greatly reduced, but at the expense of increasing the execution time. Memory requirements and execution time were further reduced by exploiting the various symmetry rules involved in computing the B_k^{lm}. The most useful rule allowed us to avoid calculating and storing the B_k^{lm} when they were zero; this is true when l is odd and k is even, or when l is even and k is odd. We were further able to reduce storage space by exploiting the fact that, while the B_k^{lm} are complex quantities, they are either purely real (if m is even) or purely imaginary (if m is odd). Finally, only the positive m values need be computed, since the negative m values involve only a sign change. By using all of these rules, we were able to reduce the storage requirements for $l = 250$ from 5.3 Mb to about 60 kb, a reduction of nearly two orders of magnitude.

When the Dilts algorithm is used for values of $l > 16$, overflows occur due to the large factorial terms in the evaluation of C_{lm} and a_{l-2j}^{ll}. To cure this, new recursion relations were developed that incorporated the factorial calculation in the computation of C_{lm} into the recursion relations for the a_k^{lm}, thereby producing direct recursion relations for the B_k^{lm} and eliminating the overflows.

The new recursion relations are as follows. For $l \neq k$ or $l \neq m$,

$$B_k^{l+1,m} = D_{lm}\,\tfrac{1}{2}[B_{k-1}^{lm} + B_{k+1}^{lm} + i(l+m)\,E_{lm}(B_{k-1}^{l,m-1} - B_{k+1}^{l,m-1})]. \tag{12}$$

The relation for the $l = k$ boundary is

$$B_l^{l+1,m} = D_{lm}\,\tfrac{1}{2}[B_{l-1}^{lm} + i(l+m)\,E_{lm}B_{l-1}^{l,m-1}] \tag{13}$$

where

$$D_{lm} = \frac{C_{l+1,m}}{C_{lm}} = \left[\frac{(2l+3)(l-m+1)}{(2l+1)(l+m+1)}\right]^{1/2} \tag{14}$$

and

$$E_{lm} = \frac{C_{lm}}{C_{l,m-1}} = \left[\frac{1}{(l+m)(l-m+1)}\right]^{1/2}. \tag{15}$$

Finally, the relation for the $l = m$ boundary is

$$B_{l+1-2j}^{l+1,l+1} = \frac{i(2l+2)}{4(l+1-j)}\left[\frac{2l+3}{2l+2}\right]^{1/2} B_{l-2j}^{ll}. \tag{16}$$

All of the factorials have been eliminated in these equations, allowing coefficients for high values of l to be computed without overflow. The seed values must now be multiplied by the appropriate values of C_{lm} to initialize the recursion. Hand calculation of a few values and comparison with Table I of the Dilts paper shows that these new recursion relations are correct.

3. Timing Tests

After solving the problems of storage and overflows, timing tests of the new B_k^{lm} generating program were performed on a VAX 8600. The results showed that the CPU time required to compute all of the B_k^{lm} up to a maximum degree L was proportional to L^3. When the modified program was run for an L of 250, the CPU time required was approximately 500 s. This time does not include the time required for the two-dimensional image interpolation and FFT, nor does it include the time required to evaluate Eq. (11) for the F_{lm}. The slow execution time of the new B_k^{lm} routine lies in the large number of operations required to evaluate the recursion relations for high l. For the new B_k^{lm} recursion relations, the number of operations required is approximately $75L^3$.

Of even more consequence is the number of operations involved in evaluating (11), given by Dilts as $40lN$. Note, however, that this number of operations is required to evaluate (11) for a *single* mode with a particular choice of l and m. If one desires to compute F_{lm} for all available modes in the image, then $L \approx N$ and, since there are a total of $L(2L+1)$ modes, the operation count to evaluate (11) for all modes becomes $40L^2(L)(2L+1) = 80L^4 + 40L^3$. The total number of operations in the Dilts algorithm to compute the spherical harmonic coefficients for every mode up to L using an $N \times N$ image is

$$n_f N^2 + n_{\mathrm{FFT}} N^2 \log N + 80L^3 N + 40L^2 N + 75L^3, \tag{17}$$

where n_f is the number of operations required to interpolate the data f onto an evenly spaced grid, and n_{FFT} is a constant reflecting the number of operations in the FFT. Typically $n_f \approx 20$ and $n_{\mathrm{FFT}} \approx 25$. If all of the modes available in the image are desired, then the number of operations will be dominated by the $80L^4$ term. When the sums in Eq. (11) were tested for L of 250, the CPU time required was about 10 h. This renders the procedure impractical for helioseismology, but it still remains viable for applications where only low values of l need be computed.

4. Conclusion

We have modified and evaluated the Dilts [1] algorithm for spherical harmonic transforms as applied to helioseismology. While we have overcome the problems of storage and overflow, the algorithm proves to be far too slow when extended to high values of l. The current best method appears still to be that first implemented by Brown [2], wherein first the image is interpolated from a $(\sin\theta, \sin\phi)$ grid to a $(\sin\theta, \phi)$ grid. Next, a one-dimensional FFT is performed in the ϕ coordinate, immediately producing the azimuthal, or m-dependence of the spherical harmonic coefficients. Finally, the associated Legendre functions P_l^m are generated using recursion relations in the $(\sin\theta)$ direction, and the data is projected onto these functions by multiplication and a one-dimensional integration. There are different strategies for generating the P_l^m [4], but with the current one in use at NSO, we can perform a spherical harmonic transform, including the interpolation and FFT for all l values and every other m value up to 250 on an 244×192 image in 50 CPU s on the VAX 8600. This is much more acceptable than the 10 CPU h per image that would be spent using the Dilts algorithm, and allows the inference of internal solar structure (e.g., [5]) to be accomplished much more efficiently.

Acknowledgments

We would like to thank Gary Dilts for useful correspondence and conversations. Mark Elowitz was supported by the National Solar Observatory Summer Research Student Program, with funding provided by the Global Oscillation Network Group.

References

1. G. A. Dilts, *J. Comput. Phys.* **57**, 439 (1985).
2. T. M. Brown, *Nature* **317**, 591 (1985).
3. J. W. Leibacher, R. W. Noyes, J. Toomre, and R. K. Ulrich, *Sci. Amer.* **253**, No. 3, 48 (1985).
4. K. G. Libbrecht, *Solar Phys.* **99**, 371 (1985).
5. T. L. Duvall, Jr., J. W. Harvey, and M. A. Pomerantz, *Nature* **321**, 500 (1986).

Received: December 12, 1986; revised: April 27, 1988

Mark Elowitz
Frank Hill

National Solar Observatory
National Optical Astronomy Observatories
950 N. Cherry Avenue
Tucson, Arizona 85726

Thomas L. Duvall, Jr.

Laboratory for Astronomy and Solar Physics
NASA/Goddard Space Flight Center
Greenbelt, Maryland 20771