Local helioseismology in the SDO HMI/AIA data analysis pipeline

R.S. Bogart1,*

Hansen Experimental Physics Laboratory, Stanford University, Stanford, CA 94305-4085, USA

Received 3 Dec 2006, accepted ?? ??? 200?
Published online later

Key words Local helioseismology data processing – Solar Dynamics Observatory

Local helioseismology techniques will play a principal role in the pipeline processing of data from HMI to produce standard data products suitable for scientific analysis. Many of these techniques are undergoing rapid development, and it is expected that new or enhanced algorithms and models will be contributed by members of the community. We describe the design and implementation of the data analysis environment of the SDO Joint Science Operations Center for the benefit of potential contributors and users.

1 Introduction

The Solar Dynamics Observatory (SDO) scheduled to be launched in late 2008 will include high-resolution (1 arcsec) full-disc imaging instruments for two of its three experiments: the Helioseismic and Magnetic Imager (HMI) for the photosphere and the Atmospheric Imaging Assembly (AIA) for the corona. Data from both of these experiments will be collected, processed, archived, analyzed, and distributed by a Joint Science Operations Center (JSOC) headquartered at Stanford University.

Local helioseismology will be a key part of the science planned for HMI. The basic elements of the planned pipeline processing and data products are discussed by Kosovichev (2007). The data products resulting from local helioseismology analysis include ring-diagram, time-distance, and acoustic holography techniques. Many, but not all, of the anticipated local helioseismology pipeline elements have already been developed in the analysis of MDI, GONG+, TON, and Mt. Wilson data. The processing procedures are however in greatly varying states of readiness for incorporation in an automated pipeline, and none are yet ready to work in the data architecture of the JSOC. We describe that architecture, presenting the requirements it imposes on pipeline module developers and its capabilities for supporting data analysis. We also present a brief review of the current status and needed future development of the various local techniques.

2 JSOC architecture

The JSOC architecture consists of three principal components: a Data Record Management System (DRMS) for the storage and retrieval of information on all data, including ancillary data, in a relational database; a Storage Unit Management System (SUMS) for the physical storage and retrieval of large volumes of data; and an Applications Programming Interface (API) to facilitate interaction with DRMS and SUMS by analysis modules to be run in the pipeline (and outside as well), together with the analysis modules themselves and supporting libraries and utilities. There is also a Pipeline User Interface to control the flow of the automatic pipeline processing of AIA and HMI data from receipt of telemetry to the standard data products, but that is not of concern here.

2.1 DRMS

The JSOC Data Record Management System is built around a relational (Postgres) database containing the JSOC data catalog. In the DRMS, data are organized into various data series of like data, each represented by a uniquely-named table in the database. Each row in a data series table corresponds to a data record, the atomic data object in the catalog. The columns of the table represent the set of (common) keys associated with each data record, and the values in each column of a given row represent the values associated with the corresponding key for the given record. A data record may include one or more named data segments, references to n-dimensional data arrays (e.g. images, data cubes) stored in files outside the table. It is possible to have columns in a data series table to link to columns in other tables, so that the value of the link in a given cell (column and row) of a table is the same as the value of that of the cell in the target table to which it is linked. This allows for propagation of information to records in dependent data series. It is also possible to have data segments associated with the records in one data series linked to those of another series.

Data records may be selected explicitly by unique identification numbers, or more typically through SQL queries...
resulting in matching of one or more key values. A data set is defined as an arbitrary set of data records selected from one or more data series. An example would be the set of HMI Dopplergrams with Observation Times within a designated interval and having a Data Quality parameter exceeding a given threshold. Pipeline modules are designed to operate on data sets as input, and may result in the creation of new data records. Data records in the DRMS are never removed; when the data in a record are to be modified, a new record is created. For this purpose, each data series has one or more primary keys defined. All records with the same union of primary key values are assumed to refer to the same data object, and the one with the highest identification number will be selected in a match for the primary key(s). In this way, “header” information may be modified without having to rewrite the data segment.

The DRMS is designed so that it can be replicated in whole or in part at different sites. Both master and slave tables are implemented at the JSOC to provide robustness, and tables from other databases can be selectively mirrored on an individual basis, with different databases serving as the master for different tables. For this purpose the table name space is managed with uniquely-assigned prefixes. For example, all standard data products of the AIA and HMI missions (and only those products) are in data series named aia.* and hmi.*, respectively. We are using the Slony-I system for mirroring and backing up of the JSOC databases. A remote mirror is currently under development at Mullard Space Science Laboratory.

The SUMS is based on a combination of dedicated disk space and tapes in a robotic system. SUMS controls the JSOC data storage resources. It manages the disk space available for storing data and the tape systems, and it tracks the current on-line location of data. SUMS is implemented as a server program and management utilities. The SUMS server uses tables in the same JSOC database as the DRMS to track “Storage Units”. A Storage Unit is simply a directory and its subdirectories, if any.

SUMS gives each managed storage unit a SUMS identification in the form Dxxxxx where xxxx is a unique serial number. SUMS allocates disk space on one of its file systems, creates a directory named with the unique identifier and SUMS identifier to the requesting program. The requesting program tells SUMS how long to keep the data online and whether or not to archive them to tape. The SUMS disks at the JSOC are configured in a RAID system and should provide secure storage online so that only the most valuable or largest collections of data need be archived to tape. SUMS also provides mechanisms for grouping data from related data series on a well-defined set of tapes.

The SUMS architecture is designed to be distributed, like the DRMS. Each DRMS instance must have its own SUMS, though it can be as small as a single dedicated disk partition. It should be possible for cooperating DRMS systems to share information about data in selected series, and to make available data that happen to be cached online at one location to another location over the network if that would be faster than staging the data from tape at the archive.
there is a documented API for direct interface to SUMS, but it is probably not of interest to module writers; it is used internally by applications involving the DRMS API. Documentation for the DRMS API is in progress.

### 2.3 DRMS API and analysis modules

Data in the DRMS may be read and written with direct `psql` commands (and proper permissions of course). It is expected though that application programs, also known as analysis modules, that require interaction with the DRMS will do so through a managed socket-level communication interface with the database. For this purpose, a library of C-language functions has been made available. There are currently 237 such functions in the API; bindings for other high-level languages are planned but not yet implemented.

In order to use the standard interface to the DRMS embodied in the API, C-language analysis modules must be implemented as functions (named `DoIt()`) to be linked to a common `main()` that initiates and manages the communication with the DRMS, passing on the environment and calling parameters directly to the module.

Because the DRMS/SUMS system is designed to support temporary, non-archived data sets of arbitrary cache lifetime, it may be desirable to isolate the many different steps of an analysis sequence (e.g. detrending, tracking, domain transformation, filtering, model fitting, inversion) as distinct modules, using temporary intermediate data products. Such intermediate products may also be shared among different possible branches of one or more pipelines.

### 3 Local helioseismology modules

The present development status of the various local helioseismology techniques for producing the expected HMI data products is summarized in Table 1. A ‘3’ means that the code used in the MDI and GONG pipelines is mature and ready to be ported to JSOC with minimal modifications. A ‘2’ means that codes exist, but may require some significant modifications or enhancements at various points to function in a pipeline. A ‘1’ means that significant work remains to be done to demonstrate that the technique will be able to reliably generate the desired data product; a ‘0’ means that it is unlikely that the technique will be applicable.

<table>
<thead>
<tr>
<th>Data Product</th>
<th>RD</th>
<th>TD</th>
<th>AH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full-disc $v, c$: 0–30 Mm</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Synoptic $v, c$: 0–30 Mm</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>High-resolution $v, c$: 0–30 Mm</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Deep-focus $v, c$: 0–200 Mm</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Farside activity index</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

There is a need to correct for solar rotation when constructing the Dopplergrams from sequential HMI filtergrams, we expect that the Dopplergrams will already be corrected for image distortion and mapped to a well-defined scale. This will obviate the need for such corrections in the mapping/tracking modules that precede most local analysis. Likewise, the production of the Dopplergrams and continuum photograms as part of the pipeline will require automation of the kinds of quality assessment that are a laborious prelude to both local and global analysis, at least in the MDI pipeline. For our purposes, the full-disc Dopplergram series of HMI data will be more nearly comparable to the merged GONG data in the degree of upstream processing.

The MDI `fastrack` module for tracking and remapping multiple target regions in parallel from a sequence of images can be ported to JSOC with little difficulty. We plan to implement an improved interpolation scheme, and questions of projections, scales, target sizes and distributions, and tracking rates must be answered by the different local helioseismology teams for the different pipeline data products. Also, a more systematic and unified approach to detrending and gap-filling is in order for all of the local helioseismology pipelines.

#### 3.2 Ring diagrams

Ring-diagram analysis for the near-surface flows in both full-disc and synoptic contexts is implemented, with a mixture of automated and manual processing, in both the MDI and GONG pipelines. Inversions for sound speed have been demonstrated in a research context, but additional work will be required to be able to produce meaningful full-disc or synoptic maps. Likewise, there is ongoing research into the feasibility of extending the depths of flows to the target of 30 Mm and below, and into achieving high spatial resolution for the near-surface flows and sound speeds. Whether ring-diagram analysis can reach the deep-focus depth target at all is doubtful.

There are currently two different approaches being used for fitting mode parameters to the power spectra, those described in Haber et al. (2002) and Basu et al. (2004). The HMI pipeline will likely incorporate some sort of hybrid, possibly borrowing techniques and/or results from high-$\ell$ global-mode ridge fitting. The quality and reliability of the inversions may likely improve from the development of inversion kernels appropriate to the particular geometry.

#### 3.3 Time distance

Time-distance analysis has not been implemented as a regular part of either the MDI or GONG pipelines, but there is sufficiently mature manual code developed to assure that the pipelines for full-disc, synoptic, and high-resolution flows and sound speed can be developed. There is some question
as to whether the deep-focus targets can be systematically achieved, so further exploratory work will be required.

One of the major outstanding issues in time-distance analysis is the selection of appropriate kernels for the inversions (cf. Couvidat et al., 2003). An early application of the DRMS could be building up a library of inversion kernels for testing and development of the ultimate time-distance analysis pipeline, including evaluation of possible forks in the procedure.

3.4 Acoustic holography

The only holographic technique so far implemented as pipeline analysis in either MDI or GONG is the production of the farside activity index. That code is fully integrated into both pipelines using essentially identical techniques, and requires only porting to be incorporated into the HMI pipeline. The production of diagnostic data products for acoustic holography (e.g. acoustic power maps) should be straightforward; such modules have already been developed in the MDI pipeline. The ability of the techniques to produce the kinds of reliable physical inferences embodied in the targeted HMI data products, such as maps of sound-speed profiles, must be demonstrated prior to their incorporation in the pipeline.

References

Kosovichev, A.G.: 2007, AN, this volume