MESSENGER X-Ray Spectrometer (XRS) EDR-to-CDR-to-RDR Processing

Version 2.1

5 January 2017

Prepared by
Gary Gindlin, Mike Fitzgibbon
The University of Arizona
Tucson, AZ 85721
USA

Table of Contents

1 Change Log

DATE	SECTIONS CHANGED	REASON FOR CHANGE	REVISION
1/7/2010	7, 9	Type of engineering smoothing was changed. Calculations of FOV data were changed.	1.2
5/29/2013	3, 14, 15	Add information regarding creation of maps of element ratios.	1.3
6/15/2013	14	Minor edits.	1.4
4/8/2014	7	Commented on the replacement of statistical outliers in SC_ANGLE and SC_RANGE values.	1.5
12/21/2015	14, 15	Updated to reflect final RDR map product generation.	1.6
1/6/2016	All	Minor edits.	1.7
1/7/2016	All	Final mission edits.	1.8
8/8/2016	3, 14, 16	Updated to include new RDR Footprint data products.	1.9
8/29/2016	3	Minor edits.	2.0
1/5/2017	14	Edits resulting from RDR Footprint peer review.	2.1

2 Purpose

This document provides a description of the conversion of MESSENGER X-Ray Spectrometer (XRS) Experiment Data Records (EDRs) to Calibrated Data Records (CDRs) and CDRs to Reduced Data Records (RDRs).

3 Introduction

The EDRs are the raw data records used to derive XRS data used for scientific and engineering analysis. The EDRs contain the Mission Elapsed Time (MET) and the raw engineering and scientific data. Before the engineering data can be used for engineering analysis, the raw data must be converted to physical units and the data must be transformed into meaningful physical reference systems. Before the science data can be used for scientific analysis, the raw data must be combined with SPICE kernel file data to create derived data. These conversions yields calibrated data which are stored in Calibrated Data Records or CDRs. The processing steps from the EDR to the CDR level are described in this document and include:

1. Convert engineering data from engineering units to physical units for each MET.
2. Smooth engineering data for each MET.
3. Derive the spatial data for each MET.
4. Derive illumination data for each MET.
5. Derive field of view data for each MET.
6. Derive downlink status for each MET.
7. Derive live time for each MET.
8. Set valid channel high and low for each MET.
9. Set real gain for each MET.
10. Set real zero for each MET.

In the sections describing the calculations of the CDR values the names of the CDR columns described in the CDR/RDR XRS SIS (Nittler, 2013) are capitalized. The calculations are shown in pseudo-code.

The RDR Footprint products are the perimeters of areas of Mercury's surface, stored in latitude and east longitude, that correspond to the portion of the surface in the instrument's field of view during integration for CDR records with FOV_STATUS of 1 or 3.

The RDR Map products are maps of the ratios of five elements (Al, Ca, Fe, Mg, S) to a sixth (Si).

4 Coordinate Systems

There are two coordinate systems in use:

- The celestial reference system used for target and spacecraft position and velocity vectors and camera pointing.
- The planetary coordinate system for geometry vectors and target location.

The celestial coordinate system is J2000 (Mean of Earth equator and equinox of J2000). The planetary coordinate system is **planetocentric**.

The following describes the computational assumptions for the geometric and viewing data provided in the PDS label:

> The mid-point time of observation is used for the geometric element computations. (The mid-point is calculated using the Start and End times from the EDR set.) Label parameters reflect observed, not true, geometry. Therefore, light-time and stellar aberration corrections are used as appropriate. The inertial reference frame is J2000 (also called EME2000). Latitudes and longitudes are planetocentric. The "sub-point" of a body on a target is defined by the surface intercept of the body-to-target-center vector. This is not the closest point on the body to the observer. Distances are in km, speeds in km/sec, angles, in degrees, angular rates in degrees/sec, unless otherwise noted. Angle ranges are 0 to 360 degrees for azimuths and local hour angle. Longitudes range from 0 to 360 degrees (positive to the East). Latitudes range from -90 to 90 degrees. SPICE kernel files are used in the geometric parameters.

5 SPICE Kernels

The MESSENGER project adopted the SPICE information system to assist science and engineering planning and analysis. SPICE is developed by the Navigation and Ancillary Information Facility (NAIF) under the directions of NASA's Science Directorate. The SPICE toolkit is available in FORTRAN and C at the NAIF web site (http://naif.jpl.nasa.gov). The MESSENGER XRS CDR processing routines are written exclusively in C using the toolkit provided by NAIF.

The primary SPICE data sets are kernels. SPICE kernels are composed of navigation and other ancillary information structured and formatted for easy access. SPICE kernels are generated by the most knowledgeable technical contacts for each element of information. Definitions for kernels include or are accompanied by metadata, consistent with flight project data system standards, which provide pedigree and other descriptive information needed by prospective users.

The following SPICE kernel files have been used to compute the UTC time and geometric quantities for the MESSENGER XRS instrument and are archived at the PDS NAIF node in the MESSSP_1000 archive volume:

1. MESSENGER spacecraft and planetary ephemeris file, also known as the planetary spacecraft ephemeris kernel (SPK) file (*.bsp).

2. MESSENGER spacecraft orientation file, also known as the attitude c-kernel (CK) file (*.bc).
3. MESSENGER reference frame files, also known as the frames and dynamic frames kernels (*.tf). The frames kernel contains the MESSENGER spacecraft, science instrument, and communication antennae frame definitions. The coordinate system required for scientific analysis is defined in the dynamic frames kernel.
4. MESSENGER instrument kernel (I-kernel) files (*.ti). The XRS instrument kernel contains references to mounting alignment, operating modes, and timing as well as internal and field of view geometry for the MESSENGER XRS. The radio science (RS) instrument kernel contains references to mounting alignment, operating modes, and timing as well as internal and field of view geometry for the MESSENGER radio science.
5. MESSENGER spacecraft clock coefficients file, also known as the spacecraft clock kernel (SCLK) file (*.tsc).
6. Planetary constants file (*.tpc), also known as the planetary constants kernel (PCK) file.
7. NAIF leap seconds kernel (LSK) file (*.tls). This kernel is used in conjunction with the SCLK kernel to convert between Universal Time Coordinated (UTC) and MESSENGER Mission Elapsed Time (MET).

An automated procedure was used during the mission to identify the most current kernel set for use in product generation. The final XRS products were generated using the final MESSENGER SPICE kernels archived at the NAIF node.

6 Conversion of Raw Engineering Data to Physical Units

The raw engineering data from the EDRs are converted into values associated with physical units (temperature, voltage, current, etc.). The converted engineering values were stored in a database table as shown in table 1 that was used in product generation and for MESSENGER team use during the mission.

Table 1 Engineering table format

Data Column Name	Data Column Description
MET	Mission elapsed time
UTC	UTC of MET
raw_val	Raw engineering data received from MESSENGER.
eng_val	The calibrated value of the reading.
smoothed_val	The smoothed value of the eng_val

The following table, table 2, has the equations used to convert the raw engineering data into physical units:

Table 2 Engineering table conversion equations

Calibrated Engineering Column Name	Equation for Converting Raw Engineering Data into Physical Units
SC_RANGE	30 * SC_RANGE
SC_ANGLE	.25 * SC_ANGLE
LVPS_PLUS_5V	.07935 * LVPS_PLUS_5V
LVPS_MINUS_5V	-.07935 * LVPS_MINUS_5V
LVPS_PLUS_12V	.07935 * LVPS_PLUS_12V
LVPS_MINUS_12V	-.07935 * LVPS_MINUS_12V
LVPS_PLUS_5_I	7.808 * LVPS_PLUS_5_I
LVPS_MINUS_5_I	7.808 * LVPS_MINUS_5_I
LVPS_PLUS_12_I	7.808 * LVPS_PLUS_12_I
LVPS_MINUS_12_I	7.808 * LVPS_MINUS_12_I
LVPS_TEMP	-39.37 + .4227 * LVPS_TEMP + -.0000449 * LVPS_TEMP ** 2 + .00000508 * LVPS_TEMP ** 3
LVPS_PRIMARY_I	7.808 * LVPS_PRIMARY_I
LVPS_SWITCHED_PRIMARY_I	7.808 * LVPS_SWITCHED_PRIMARY_I
GPC1_MG_PLUS_5V	.0421 * GPC1_MG_PLUS_5V
GPC2_AL_PLUS_5V	.0421 * GPC2_AL_PLUS_5V
GPC3_UN_PLUS_5V	.0421 * GPC3_UN_PLUS_5V
SAX_PLUS_5V	.0421 * SAX_PLUS_5V
ANALOG_PLUS_5V	.0421 * ANALOG_PLUS_5V
DIGITAL_PLUS_5V	.0421 * DIGITAL_PLUS_5V
GPC1_MG_MINUS_5V	.02559 * GPC1_MG_MINUS_5V + -068202.0 * GPC1_MG_PLUS_5V
GPC2_AL_MINUS_5V	.02559 * GPC1_MG_MINUS_5V + -068202.0 * GPC1_MG_PLUS_5V
GPC3_UN_MINUS_5V	.02559 * GPC2_AL_MINUS_5V + -068202.0 * GPC2_AL_PLUS_5V

SAX_MINUS_5V	.02559 * GPC3_UN_MINUS_5V + -068202.0 * GPC3_UN_PLUS_5V
ANALOG_MINUS_5V	-12.1 + .05732 * ANALOG_MINUS_5V
TEC_I	2.34 * TEC_I
MXU_TEMP	MXU_TEMP * (-26.226 * log(MXU_TEMP + 1) + 129.14)

SOLAR_DETECTOR_TEMP	If (PIN_TEC_MODE == 1) Then
	/* high temperature (anneal) */
	.00000000457782 * SOLAR_DETECTOR_TEMP ** 5 +
	-.00000316578 * SOLAR_DETECTOR_TEMP ** 4 +.000858411 * SOLAR_DETECTOR_TEMP ** 3 -.111961 * SOLAR_DETECTOR_TEMP ** 2 + 7.16858 * SOLAR_DETECTOR_TEMP +-123.365
	Else
	/* low temperature (normal) */
	2.06686 * log(SOLAR_DETECTOR_TEMP + 1) ** 2 + -38.94592 * log(SOLAR_DETECTOR_TEMP + 1) +107.39573

SAX_TEMP	-273 + 1.47 * SAX_TEMP
SOLAR_DETECTOR_I	-667 + 4.017 * SOLAR_DETECTOR_I
GPC1_MG_VOLTAGE	.507 * GPC1_MG_VOLTAGE
GPC2_AL_VOLTAGE	.507 * GPC2_AL_VOLTAGE
GPC3_UN_VOLTAGE	.507 * GPC3_UN_VOLTAGE
BIAS_VOLTAGE	.507 * BIAS_VOLTAGE
GPC1_MG_SUPPLY_TEMP	-99.4 + 1.028 * GPC1_MG_SUPPLY_TEMP
GPC2_AL_SUPPLY_TEMP	-101.4 + 1.028 * GPC2_AL_SUPPLY_TEMP
GPC3_UN_SUPPLY_TEMP	-100.4 + 1.028 * GPC3_UN_SUPPLY_TEMP
BIAS_SUPPLY_TEMP	-98.3 + 1.028 * BIAS_SUPPLY_TEMP

7 Smoothing of Engineering Data

The engineering smoother process detects and replaces anomalous data points from the XRS engineering data. It does not actually perform a traditional smoothing to remove noise (Marchand and Marmet, 1983). It is simply a tool for the detection and replacement of points that are statistical outliers. The engineering smoother process uses the “outliers” library from the R statistical package for outlier detection.

For each engineering channel four parameters are defined.

1. A search window size to look for outliers
2. A mean window size to generate replacement values
3. A statistical scoring method
4. A score threshold that separates the outliers

Table 2 shows the default values used for all engineering channels.

Table 2: Engineering smoother parameters

Name	Value
Search N	50
Mean N	50
Method	Z
Threshold	5.0

Note that the actual search and mean windows include 2*N+1 points.

The algorithm proceeds by first calculating the "z" normal scores (differences between each value and the mean divided by the standard deviation) for every point in the search window. Then the results are checked to determine if the absolute value of the score at the window center exceeds the threshold. If so that point is replaced with the mean value of all points in the mean window which are not also outliers.

In the source EDR data, SC_ANGLE and SC_RANGE are 2-byte integer arrays. The conversion factor for SC_RANGE is 30 meters, so its maximum valid value is 1966080 meters or 1966 km. Because of MESSENGER's highly elliptical orbit, most SC_RANGE (and corresponding SC_ANGLE) values are out of range. Out of range and erroneous SC_ANGLE and SC_RANGE values are indicated by a value of "-1" in the EDR data. In the corresponding CDR products covering data collected up through September 17, 2013, SC_ANGLE and SC_RANGE values that were not marked as "-1" in the EDR data, but were surrounded by values of "-1" in the same sampling window, were sometimes incorrectly determined to be statistical outliers and were replaced with an average of the non-outlier values which would result in a value of "-1" as well. Outlier substitution is not done in products covering data collected after September 17, 2013. In subsequent products and in earlier products that have been reprocessed since March 12, 2014, the SC_ANGLE and SC_RANGE values contain values that are converted from raw units without the replacement of statistical outliers. Processing and reprocessing dates are indicated in the PRODUCT_CREATION_TIME field in the associated PDS labels.

8 Spatial Data Calculations

The following spatial data is calculated:

ANGLE_SUN_SOLAR_MONITOR
BS_VECTOR_[X,Y,Z]
DELTA_ANGLE
EARTH_POSITION_[X,Y,Z]
INSTRBORESIGHT_MERCURY
INTERSECTION
HERMEOCENTRIC_LATTITUDE
HERMEOCENTRIC_LONGITUDE
LOCAL_HOUR
LOCAL_MINUTE
MERCURY_SOL
PHASE_ANGLE
POINTING
SCALT
SC_POSITION_[X,Y,Z]
SUN_DISTANCE
SUN_POSITION_[X,Y,Z]

These calculations are performed for the j2000 value of the MIDPOINT_MET. The conversion from the MIDPOINT_MET to j2000 is handled by the SPICE toolkit using the following code:

```
partition = 1
sprintf(sclkch, "%d/%d", partition, MIDPOINT_MET);  // This creates a character string  with
                                                    the partition number followed by '/'
                                                    followed by the MIDPOINT_MET
scs2e_c(sc_id, sclkch, j2000)
```

The routine scs2e_c converts the MIDPOINT_MET into ephemeris time j2000, whereby the spacecraft ID (sc_id) for MESSENGER is -236.

8.1 SCALT Calculation

Find the sub-spacecraft point of MESSENGER on the surface of Mercury using the "Intercept" method (point on the surface of Mercury on a line between MESSENGER and the center of Mercury) and find the spacecraft altitude, SCALT. This is handled by the SPICE toolkit using the following code:

```
subpt_c("Intercept", "Mercury", j2000, "NONE", "MESSENGER",
        spoint, SCALT)
```

The returned value spoint is an array with the sub-spacecraft point on the surface of Mercury at time of j2000 and the returned value SC_ALT is a reference to the altitude of MESSENGER at the time of j2000.

8.2 HERMEOCENTRIC_LATITUDE and HERMEOCENTRIC_LONGITUDE Calculation

Convert the sub-spacecraft point of MESSENGER on the surface of Mercury into latitude and longitude along with the radii of Mercury. This is handled by the SPICE toolkit using the following code:

```
reclat_c(spoint, mercury_radii, HERMEOCENTRIC_LONGITUDE,
        HERMEOCENTRIC_LATITUDE)
```

where spoint is the array with the point on the surface of Mercury returned in the subpt_c call from the SCALT calculation above.

The returned value mercury_radii is an array with the radii of Mercury, the returned value HERMEOCENTRIC_LONGITUDE is a reference to the longitude of Mercury at the point on the surface of Mercury under MESSENGER and the returned value HERMEOCENTRIC_LATITUDE is a reference to the latitude at the point on the surface of Mercury under MESSENGER.

The longitude and latitude are converted to degrees from radians by multiplying them by 180.0 and dividing that product by PI.

8.3 POINTING Calculation

Check if a SPICE CK kernel file, the SPICE kernel with spacecraft orientation information, contains data for the j2000 value.

For performing this, convert the j2000 value to spacecraft clock (‘ticks’). The conversion from the j2000 to ticks is handled by the SPICE toolkit using the following code:

```
sce2t_c(sc_id, j2000, ticks)
```

The routine sce2t_c converts the j2000 value into spacecraft clock (‘ticks’), whereby the spacecraft ID for MESSENGER is -236.

The orientation for of MESSENGER for the spacecraft clock is obtained by calling the SPICE ck kernel get pointing info api as follows:

```
ckgp_c(sc_ck_id, ticks, 0.0, “J2000”, matrix, clckout, found)
```

where sc_ck_id is the spacecraft CK SPICE kernel id of -236000.

If the orientation data exists for the spacecraft clock then the following step occur:

POINTING is set to 1.

If the orientation data does not exist for the spacecraft clock then the following step occur:

POINTING is set to 0.

8.4 INSTRBORESIGHT_MERCURY, INTERSECTION Calculation

If the SPICE orientation data exists at time j2000 as determined in the previous step then perform the following steps:

1. Get the position of MESSENGER as seen from Mercury in the inertial frame, J2000. This is handled by the SPICE toolkit using the following code:

   ```
   spkezr_c("MESSENGER", j2000, "J2000", "NONE", "MERCURY",
            state, lt)
   vequ_c(state[0], scpos_body)
   ```

2. Get the transformation matrix from the instrument frame, "MSGR_GRNS_GRS", to the body-fixed rotating frame, "J2000", at time j2000 using the following code:

   ```
   pxform_c("MSGR_GRNS_GRS", "IAU_MERCURY", j2000, instr2body)
   ```

3. Find the instrument "boresight" direction to the body-fixed frame using the following code:

   ```
   vpack(0.0, 0.0, 1.0, bsight_instr)
   mxv_c(instr2body, bsight_instr, bsight_body)
   ```

4. Find the INSTRBORESIGHT_MERCURY with the following code:

   ```
   surfpt_c(scpos_body, bsight_body, mercury_radii[0], mercury_radii[1],
            mercury_radii[2], INSTRBORESIGHT_MERCURY, found)
   ```

   where scpos_body is first three elements in the state array returned in the call to spkezr_c from a an earlier step.

5. If the INSTRBORESIGHT_MERCURY is found then INTERSECTION is set to 1 else INTERSECTION is set to 0.

6. If the SPICE orientation data for time j2000 does not exist then INSTRBORESIGHT_MERCURY is set to an array of 0.0, 0.0, 0.0 and INTERSECTING is set to 0.

8.5 DELTA_ANGLE Calculation

Find the DELTA_ANGLE using the following steps:

If the INSTRBORESIGHT_MERCURY is found then

1. Find the radii of Mercury:

   bodvar_c(199, "RADII", n, radii)

   where 199 is the body id of Mercury.

2. Find the normal to the Mercury at the surface:

   surfnm_c(radii[0], radii[1], radii[2], INSTRBORESIGHT_MERCURY, normal)

3. Create a vector representing north using the following code:

   vpack(0.0, 0.0, 1.0, north)

4. Find the transformation matrix, body2local, having the normal to the surface of Mercury as a +Z axis and having the north vector lying on the X plane with the following code:

   twovec_c(normal, 3, north, 1, body2local)

5. Create vector representing instrument reference direction, +Y axis of instrument head frame:

   vpack(0.0, 1.0, 0.0, refdir_instr)

6. Rotate the instrument reference direction to the local level frame using the following code:

   mxv_c(instr2body, refdir_instr, refdir_body)
   mxv_c(body2local, refdir_body, refdir_local)

   where instr2body and body2local are the transformation matrices returned in previous steps.

7. Convert rectangle coordinates to range, right ascension, DELTA_ANGLE, and declination:

   recrad_c(refdir_local, range, DELTA_ANGLE, dec)

   where range is a reference to the range, DELTA_ANGLE is a reference to the right ascension and dec is a reference to the declination.

   The DELTA_ANGLE is converted to degrees from radians by multiplying it by 180.0 and

dividing that product by PI.

If the INSTRBORESIGHT_MERCURY is not found then
1. Set DELTA_ANGLE to 0.0.

8.6 LOCAL_HOUR, LOCAL_MIN Calculation

If the orientation information was found in a previous step then

1. Get local hour, LOCAL_HOUR, and the local minute, LOCAL_MIN, at the longitude of the sub-point of MESSENGER on the surface of Mercury with the following code:

```
et2lst(j2000, 199, MERCURY_LONGITUDE, "Planetocentric",
        LOCAL_HOUR, LOCAL_MIN, local_second,
        localtime, local_ampm,14, 40, 5)
```

where 199 is the body id of Mercury and MERCURY_LONGITUDE is from an above step.

2. If the orientation information was not found in a previous step then LOCAL_HOUR and LOCAL_MIN are set to 0.

8.7 ANGLE_SUN_SOLAR_MONITOR Calculation

Find the ANGLE_SUN_SOLAR_MONITOR, the angle between the Sun and the solar monitor, using the following steps:

1. Get the state (position and velocity) of the SUN as seen from MESSENGER in the "MSGR_XRS_SAX" frame. This is handled by the SPICE toolkit using the following code:

```
spkez_c(10, j2000, "MSGR_XRS_SAX", "NONE", -236, state, lt)
```

where 10 is the id of the Sun, -236 is the id of Mercury and "MSGR_XRS_SAX" is the observer's frame of reference.

2. Get the angular separation between Mercury and the boresight_vector set to (0.0, 0.0, 1.0):

```
angle = vsep_c(state, boresight_vector)
```

3. Convert the angular value from radians to degrees:

```
ANGLE_SUN_SOLAR_MONITOR = angle * dpr_c()
```

where dpr_c is a SPICE call that returns the number of degrees per radian

8.8 BS_VECTOR Calculation

Find the BS_VECTOR, the boresight vector in J2000 Frame, with the following steps:

1. Get the transformation matrix from the instrument frame, "MSGR_XRS_MXU", to the body-fixed rotating frame, "J2000", at time j2000 using the following code:

   pxform_c("MSGR_XRS_MXU", "J2000", j2000, instr2body)

2. Find the instrument "boresight" direction for the boresight_vector set to (0.0, 0.0, 1.0) in the body fixed frame using the following code:

   mxv_c(instr2body, boresight_vector, BS_VECTOR)

8.9 EARTH_POSITION Calculation

Find the EARTH_POSITION, the position of the Earth in Mercury fixed coordinate system, with the following steps:

1. Get the state (position and velocity) of the Earth as seen from Mercury in the "IAU_MERCURY" frame. This is handled by the SPICE toolkit using the following code:

   spkez_c(399, j2000, "IAU_MERCURY", "NONE", 199, state, lt)

   where 399 is the id of the Earth, 199 is the id of Mercury and "IAU_MERCURY" is the observer's frame of reference.

2. Get the position from the state of the Earth:

   vequ_c(state[0], EARTH_POSITION)

8.10 MERCURY_SOL Calculation

Get the MERCURY_SOL, the longitude of the Sun, as seen from Mercury at the time j2000, in degrees using the following code:

MERCURY_SOL = dpr_c() * lspcn_c("MERCURY", j2000, "NONE")

where dpr_c is a SPICE call that returns the number of degrees per radian and lspcn_c is the SPICE call that computes the longitude of the Sun as seen from a specified body.

8.11 PHASE_ANGLE Calculation

Get the PHASE_ANGLE, the angle between direction from the Sun to the surface and direction from the surface to the spacecraft, using the following steps:

1. Translate the name of the body to the corresponding SPICE integer ID code::

bodn2c_c(“MESSENGER”, obs_id)

2. Get the state (position and velocity) of the Sun as seen from MESSENGER in the “MSGR_SPACECRAFT” frame. This is handled by the SPICE toolkit using the following code:

   spkez_c(10, j2000, “MESSENGER_SPACECRAFT”, “NONE”, obs_id,
   state_sun, lt)

where 10 is the id of the SUN, obs_id is the id of MESSENGER from the above step and “MESSENGER_SPACECRAFT” is the observer’s frame of reference.

3. Get the state (position and velocity) of Mercury as seen from MESSENGER in the “MSGR_SPACECRAFT” frame. This is handled by the SPICE toolkit using the following code:

   spkez_c(199, j2000, “MESSENGER_SPACECRAFT”, “NONE”, obs_id,
   state_mercury, lt)

where 199 is the id of Mercury, obs_id is the id of MESSENGER from an earlier step and “MESSENGER_SPACECRAFT” is the observer’s frame of reference.

4. Get the angular separation between the Sun and Mercury as seen from MESSENGER:

   angle = vsep_c(state_sun, state_mercury)

5. Convert the angular value from radians to degrees:

   PHASE_ANGLE = angle * dpr_c()

8.12 SC_POSITION Calculation

Get the SC_POSITION, the position of the spacecraft in the inertial frame, J2000, with the following steps:

1. Get the state (position and velocity) of the Earth as seen from Mercury in the “J2000” frame. This is handled by the SPICE toolkit using the following code:

   spkez_c(-236, j2000, “J2000”, “NONE”, 199, state, lt)

where -236 is the id of the MESSENGER, 199 is the id of Mercury and “J2000” is the observer’s frame of reference.

2. Get the position from the state of the spacecraft:

vequ_c(state[0], SC_POSITION)

8.13 SUN_DISTANCE Calculation

Get the SUN_DISTANCE, the Sun position vector magnitude, with the following steps:

1. Get the state (position and velocity) of the Sun as seen from MESSENGER in the "MSGR_SPACECRAFT" frame. This is handled by the SPICE toolkit using the following code:

   spkez_c(10, j2000, "MESSENGER_SPACECRAFT", "NONE", -236, state, lt)

   where 10 is the id of the Sun, -236 is the id of MESSENGER and "MSGR_SPACECRAFT" is the observer's frame of reference.

2. Get the position from the state of the Sun:

   vequ_c(state[0], position_sun_msgr_spacecraft_frame)

3. Get the distance between MESSENGER and the Sun.

   SUN_DISTANCE = vdist_c(position_sun_msgr_spacecraft_frame, bodypos_inertial)

   where bodypos_inertial is the vector (0.0, 0.0, 0.0)

8.14 SUN_POSITION Calculation

Get the SUN_POSITION, position of the Sun in the Mercury fixed coordinate system, with the following steps:

1. Get the state (position and velocity) of the Sun as seen from MESSENGER in the "IAU_MERCURY" frame. This is handled by the SPICE toolkit using the following code:

   spkez_c(10, j2000, "IAU_MERCURY", "NONE", 199, state, lt)

   where 10 is the id of the Sun, 199 is the id of MERCURY and "IAU_MERCURY" is the observer's frame of reference.

2. Get the position from the state of the Sun:

   vequ_c(state[0], SUN_POSITION)

9 Field of View Data Calculations

Several of the X-Ray spectrometer spatial columns are based on the intersection of the field of view (FOV) with Mercury, the footprint, and the terminator on Mercury, the illuminated footprint: AVG_EMISSION_ANGLE, AVG_INC_EMI_COS_RATIO, AVG_INCIDENCE_ANGLE, AVG_SC_DISTANCE,

FOOTPRINT_SOLID_ANGLE, FOV_STATUS, TOTAL_EFF_SOLID_ANGLE, TOTAL_ILLUMINATED_AREA and TOTAL_VISIBLE_AREA. The footprint is expressed as the set of HEALPix region numbers that are to the inside of all the FOV edges and on the spacecraft side of the limb. The illuminated footprint is the set of footprint HEALPix that are also on the sunward side of the terminator.

The AVG_EMISSION_ANGLE corresponds to the average emission angle within the illuminated footprint. The AVG_INC_EMI_COS_RATIO corresponds to the average ratio of the cosine of the incidence angle to the cosine of the emission angle within the illuminated footprint. The AVG_INCIDENCE_ANGLE corresponds to the average incidence angle within the illuminated footprint. The AVG_SC_DISTANCE corresponds to the average distance of MESSENGER from the surface of Mercury in the illuminated footprint. The FOOTPRINT_SOLID_ANGLE corresponds to the overall solid angle of the illuminated footprint of the X-Ray spectrometer's field of view. The FOV_STATUS is a value in the range 0-4 where 0 is for a FOV completely off Mercury (gftfov_c found no intervals), 1 is for a FOV totally on Mercury and at least part of the footprint is lit by the Sun, 2 is for a FOV totally on Mercury and the FOV is totally dark, 3 is for a FOV partially off Mercury and is partially lit by the Sun and 4 is for a FOV partially off Mercury and the FOV is totally dark. The TOTAL_EFF_SOLID_ANGLE corresponds to the total solid angle weighted by the response function of the collimator. The TOTAL_ILLUMINATED_AREA corresponds to the total area of the illuminated footprint on Mercury visible from MESSENGER. The TOTAL_VISIBLE_AREA corresponds to the total area of the footprint on Mercury visible from MESSENGER.

To determine if there is any intersection at all the SPICE geometry finder routine, gftfov_c, is used to find the interval(s) within the integration period that Mercury is within the XRS FOV. Then the only planet flag is set to true and, for each interval, the effective duration is calculated as 'interval duration (s)'/'actual integration time (s)' and the footprint sets, only planet flag, and relative areas of the FOV, full footprint, and illuminated footprint are calculated at the center of the interval. If there is no intersection all of the spatial columns calculated here are set to zero, otherwise these values are calculated for each interval using the following steps in Mercury's body-fixed frame of reference:

9.1 HEALPix calculations

The footprint HEALPix region numbers are specified as follows:

1. Each region is a bit field at its number offset within an array of bit fields.
2. A region is included in a group if the bit field for that group is set.
3. A group bit is set if the center of the region is within the group's disc.
4. A group disc is defined as the vector from center of Mercury to the center of the disc and an angle from that vector to the edge of the disc.

5. The disc edge is either the limb, the intersection of the plane defining an FOV edge and Mercury, or the terminator for the illuminated portion. If the FOV edge does not intersect Mercury, that group is ignored.

6. The disc center vector is the sub-spacecraft point for the limb, the sub-solar point for the terminator, and the normal to the FOV edge plane that points toward the boresight vector.

7. The footprint HEALPix region numbers are then the indices of all the bit fields that have the limb and all the non-ignored FOV edge group bits set and the illuminated footprint also have the terminator group bit set.

9.2 Only planet flag

The only planet flag is set to false if any of the FOV corners do not intersect Mercury.

9.3 Relative areas calculations

The limb, FOV edge intersections, and terminator are used to map out the edges of the full and illuminated footprint on the Mercury as sections of their corresponding ellipses. These sections are then mapped to points on the planet with 1 degree of arc between each point. These points, and the corners of the FOV, are then projected from the spacecraft onto the plane contain the limb. Theses sets of points describe polygons, and the areas of the polygons are directly proportional to the areas of the FOV, full footprint, and illuminated footprint.

9.4 TOTAL_VISIBLE_AREA

The TOTAL_VISIBLE_AREA is then the number of HEALPix regions within the full footprint multiplied by the area of a single region.

9.5 TOTAL_ILLUMINATED_AREA

The TOTAL_ILLUMINATED _AREA is then the number of HEALPix regions within the illuminated footprint multiplied by the area of a single region.

9.6 AVG_EMISSION_ANGLE

The AVG_EMISSION_ANGLE is then the mean of the angles between the normal to the center of each of the illuminated HEALPix regions and the vector from the center of each of the illuminated HEALPix regions to the spacecraft.

9.7 AVG_INCIDENCE_ANGLE

The AVG_INCIDENCE_ANGLE is then the mean of the angles between the normal to the center of each of the illuminated HEALPix regions and the vector from the center of each of the illuminated HEALPix regions to the Sun.

9.8 AVG_INC_EMI_COS_RATIO

The AVG_INC_EMI_COS_RATIO is then the mean of the ratios in each of the illuminated HEALPix regions, cosine(incidence angle)/cosine(emission angle).

9.9 AVG_SC_DISTANCE

First a sum of the distance from the center of each of the illuminated HEALPix regions to the spacecraft multiplied by its cosine(emission angle) is calculated. The division is to weight each region by its apparent size. The AVG_SC_DISTANCE is then the sum divide by the sum of all the cosine(emission angle).

9.10 FOOTPRINT_SOLID_ANGLE

The FOOTPRINT_SOLID_ANGLE is the FOV solid angle multiplied by the ratio (illuminated footprint area)/(FOV area) from section 8.2 and then multiplied by the effective duration as described above (third paragraph of section 8).

9.11 TOTAL_EFF_SOLID_ANGLE

For each illuminated HEALPix region an X-Ray spectrometer response is calculated from the angles between the vector from the spacecraft to the center of each of the illuminated HEALPix regions and the boresight vector. Then the sum of each response multiplied by its cosine(emission angle) is calculated. The TOTAL_EFF_SOLID_ANGLE is then the sum divided by the sum of all the cosine(emission angle).

The X-Ray spectrometer response is calculated as follows from a gaussian plus second-order polynomial fit (angle in degrees):

```
z = ((-angle) - coef[1])/coef[2]
response = coef[0] * exp(-0.5 * (z*z)) + (coef[5]*(-angle) + coef[4])*(-angle) + coef[3]
```

. where:

> The negative side of the fit is used since its zero crossing is later, -6.02 versus 6.002.
> If angle > 6.0209 then the response is zero.
> coef[6] = {1.10683, -0.00116774, 2.28958, -0.106825, -1.71824e-005, 0.00198079}

If there is more than one interval then the AVG_EMISSION_ANGLE, AVG_INCIDENCE_ANGLE, AVG_INC_EMI_COS_RATIO, AVG_SC_DISTANCE, TOTAL_ILLUMINATED_AREA and TOTAL_VISIBLE_AREA are the mean of the interval calculations weighted by the relative interval durations. The FOOTPRINT_SOLID_ANGLE and TOTAL_EFF_SOLID_ANGLE sum together the interval values.

Finally the FOV_STATUS starts at 1, since if there was no intersection it would have been set to zero. If the number of illuminated HEALPix regions is zero, then 1 is added to the FOV_STATUS. If the only planet flag is false, then 2 is added to the FOV_STATUS.

10 Live Time Calculations

The following steps are used to set the live times GPC1_MG_LIVE_TIME, GPC2_AL_LIVE_TIME, GPC3_UN_LIVE_TIME and SAX_LIVE_TIME, the actual integration times, of the GPC1_MG, GPC2_AL and GPC3_UN detectors:

```
if (gpc1_mg_center_anode_rate – gpc1_mg_veto_anode_rate > 0)
{
    GPC1_MG_LIVE_TIME =
        actual_integration_time * gpc1_mg_valid_rate/ (gpc1_mg_center_anode_rate
          gpc1_mg_veto_anode_rate)
 }
 else
 {
     GPC1_MG_LIVE= 0
 }
 if (gpc2_al_center_anode_rate – gpc2_al_veto_anode_rate > 0)
{
   GPC2_AL_LIVE_TIME =
        actual_integration_time * gpc2_mg_valid_rate/ (gpc2_al_center_anode_rate
          gpc2_al_veto_anode_rate)
}
else
{
   GPC2_AL_LIVE= 0
}
if (gpc3_un_center_anode_rate – gpc3_un_veto_anode_rate > 0)
{
   GPC3_UN_LIVE_TIME =
        actual_integration_time * gpc3_un_valid_rate/ (gpc3_un_center_anode_rate
          gpc3_un_veto_anode_rate)
}
else
{
   GPC3_UN_LIVE= 0
}
if  (solar_monitor_rate > 0)
{
   SAX_LIVE_TIME =
        actual_integration_time * solar_monitor_valid_rate/ solar_monitor_rate
}
else
{
   SAX_LIVE_TIME= 0
}
```

where the following parameters are from the XRS EDR data at the mission elapsed time:

actual_intgeration_time is the actual integration period in seconds, the gpc1_mg_valid_rate, the gpc2_al_valid_rate, gpc3_un_valid_rate and solar_monitor_valid_rate are the valid rates per integration period, the gpc1_mg_veto_anode_rate, gpc2_al_veto_anode_rate and gpc2_un_veto_anode_rate are the veto anode rate per integration period, the gpc1_mg_center_anode_rate, gpc2_al_center_anode_rate and the gpc3_un_center_anode_rate are the anode rates per integration period and the solar_monitor_rate is the solar monitor detector rate per integration period.

11 Valid Channel High and Low Calculations

The following steps are used to set the valid channel high and low values for GPC1_MG_VALID_CHANNEL_HI, the GPC1_MG filtered X-Ray rise time valid discriminator threshold, GPC1_MG_VALID_CHANNEL_LOW, the GPC1_MG filtered X-Ray lower level discriminator threshold., GPC2_AL_VALID_CHANNEL_HI, the GPC2_AL filtered X-Ray rise time valid discriminator threshold, GPC2_AL_VALID_CHANNEL_LOW, the GPC2_AL filtered X-Ray lower level discriminator threshold., GPC3_UN_VALID_CHANNEL_HI, GPC3_UN filtered X-Ray rise time valid discriminator threshold and GPC3_UN_VALID_CHANNEL_LOW, the GPC3_UN filtered X-Ray lower level discriminator threshold.. Events lower than the low values or higher than the high values are rejected.

```
GPC1_MG_VALID_CHANNEL_HI = 253.0
GPC2_AL_VALID_CHANNEL_HI = 253.0
GPC3_UN_VALID_CHANNEL_HI = 253.0
GPC1_MG_VALID_CHANNEL_LOW = 10.0
GPC2_AL_VALID_CHANNEL_LOW = 10.0
GPC3_UN_VALID_CHANNEL_LOW = 10.0
if (gpc1_mg_low_level_disc > 10.0)
{
    GPC1_MG_VALID_CHANNEL_LOW = gpc1_mg_low_level_disc
}
if (gpc2_al_low_level_disc > 10.0)
{
    GPC2_AL_VALID_CHANNEL_LOW = gpc2_al_low_level_disc
}
if (gpc3_un_low_level_disc > 10.0)
{
    GPC3_UN_VALID_CHANNEL_LOW = gpc3_un_low_level_disc
}
```

where the following are from the XRS EDR data at the mission elapsed time:
gpc1_mg_low_level_disc, gpc2_al_low_level_disc and gpc3_un_low_level_disc are low level discriminators.

12 Set Real Gain

The following steps are used to set the real gain of the GPC1_MG, GPC2_AL and GPC3_UN detectors in keV:

GPC1_MG_REAL_GAIN = 0.0383
GPC2_AL_REAL_GAIN = 0.0383
GPC3_UN_REAL_GAIN = 0.0379

13 Set Real Zero

The following steps are used to set the real zero of the GPC1_MG, GPC2_AL and GPC3_UN detectors in keV:

GPC1_MG_REAL_ZERO = 0.383
GPC2_AL_REAL_ZERO = 0.383
GPC3_UN_REAL_ZERO = 0.379

14 Footprints

The footprints represent the perimeter of the area on the planetary surface in the instrument's field of view during the integration period. Footprints are produced for CDR records collected during the main and extended science missions when the CDR record's FOV_STATUS is 1 or 3; that is, when all or a part of the field of view falls on the planet and all or part of that field of view is lit by the sun.

For each CDR record in the set, a base HEALPix map is selected at a resolution based on the midpoint altitude of the CDR observation. HEALPix creates a map on a sphere with equal area, curvilinear quadrilateral pixels centered on equally spaced latitude bands. The base map has twelve pixels that are subdivided based on the NSIDE parameter, which determines the number of subpixels placed along the side of each base map pixels; each pixel is, thus, divided into NSIDE by NSIDE subpixels. At altitudes at or below 10 km, NSIDE is set to 4096. At altitudes greater than 10 but less than and including 60 km, NSIDE is 1024. At altitudes above 60 km, NSIDE is 256. These values were determined experimentally and were modified until the consensus of the team was that the footprints produced reasonably represented the true footprint of the observations. Because the footprint perimeters are vertices of the HEALPix map at the selected resolution, the resolution of the map necessarily adds some uncertainty to the map boundaries.

For each CDR observation, the midpoint altitude and the instrument field of view are used to mark pixels on the HEALPix map that fall within the observation's footprint. After all constituent pixels are marked, a perimeter is derived whose vertices are the exterior vertices of the boundary pixels. This perimeter is stored as latitude and east longitude pairs in decimal degrees. Pole-spanning footprints may contain a small number of anomalous vertices in the interior of the footprint shape; these are an artifact of the HEALPix processing and may be ignored.

There are two suggested methods for correlating footprints to their CDR record counterparts. When processing the dataset in bulk, the spacecraft clock can be used a primary key for associating footprints and CDR records. When processing much smaller sets, the UTC date can be used as a first key to constrain the search. CDR records are organized into folders based on UTC year, month, and day. Within a single UTC day, all CDR records are collected into a single file with each observation a row in that file. Footprints are organized into folders based on UTC year, month, day, and hour. Within a single UTC hour, footprints are stored as individual files. To verify the correspondence of a footprint and CDR record, the spacecraft clock should be used.

15 Element Ratio Maps

The maps show the ratio of five elements (magnesium, aluminum, sulfur, calcium, and iron) to silicon; for each element ratio, a map of the uncertainty of the ratio and a map of effective spatial resolution are also provided. The maps are produced using the XRS spectra and the footprint calculations described in this document, following procedures described in Weider et al. (2014, 2015).

The maps are 1440 × 720 pixels corresponding to 1/4-degreee pixels in cylindrical projection centered on lat,lon=0,0. Thus, the first column represents longitude = -180 to -179.75 degrees, the bottom row is -90 to -89.75 degrees south latitude, etc. All images are 8-bit. The ratio images are linearly scaled such that the maximum pixel value (255) corresponds to a maximum map value as provided in the label. For example, if the maximum Mg/Si value is given as 0.782, then to get the actual ratio value from a pixel, one would multiply the pixel values from the Mg/Si jpeg by a factor = 0.782/255 = 0.00307. Pixels with zero value represent regions without XRS data.

16 References

Nittler, L., MESSENGER X-Ray Spectrometer Calibrated and Reduced Data Record Software Interface Specification, MESSENGER H XRS REDUCED DATA RECORD (RDR) MAPS V1.0, data set MESS-H-XRS-3-RDR-MAPS-V1.0, NASA Planetary Data System, 2013.

Marchand, P. and Marmet, L. (1983) Binomial Smoothing Filter - A Way to Avoid Some Pitfalls of Least-Squares Polynomial Smoothing. Review of Scientific Instruments. 54 (8), 1034-1041.

Weider, S.Z., Nittler, L.R., Starr, R.D., Crapster-Pregont, E.J., Peplowski, P.N., Denevi, B.W., Head, J.W., Byrne, P.K., Hauck, S.A. II., Ebel, D.S. and Solomon, S.C. (2015) Evidence for geochemical terranes on Mercury: Global mapping of major elements with MESSENGER's X-Ray Spectrometer. Earth Planet. Sci. Lett. 416, 109-120.

Weider, S.Z., Nittler, L.R., Starr, R.D., McCoy, T.J. and Solomon, S.C. (2014) Variations in the abundance of iron on Mercury's surface from MESSENGER X-Ray Spectrometer observations. Icarus. 235, 170-186.

K. M. Gorski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, M. Bartelman, HEALPix: A Framework for High-Resolution Discretization and Fast Analysis of Data Distributed on the Sphere, Astrophysics, 622, 759-771, 10.1086/427976, 2004.