

MESSENGER Neutron Spectrometer (NS) EDR-to-CDR-to-DDR Processing

Version 7.0

9 February 2016

Prepared by
David J. Lawrence
Johns Hopkins University Applied Physics Laboratory

Table of Contents

1	Purpose.....	3
2	Introduction.....	3
3	Coordinate Systems	4
4	SPICE Kernels	5
5	Time-Series Corrections to Count Rate and Spectra Data.....	6
6	Conversion of Raw Engineering Data to Physical Units.....	7
7	Derivation of Net Neutron Count Rates (NS_DDR_NCR)	9
7.1	Data Selection	9
7.2	Gain Corrections	9
7.3	Background Subtraction.....	10
8	Analysis of Net Neutron Count Rates (NS_DDR_NCP).....	12
9	Spatial and Velocity Data Calculation.....	12
9.1	SC_ALTITUDE Calculation	13
9.2	SC_LATITUDE and SC_LONGITUDE Calculation.....	13
9.3	VEL_VECTOR_J2000 and VEL_NORM Calculations.....	13
9.4	V_DOT_X Calculation	14
9.5	VEL_VECTOR Calculation	15
9.6	SC_TO_NADIR_ROT Calculation	16
10	References.....	17
11	Revision History	17

1 Purpose

This document provides a description of the conversion of MESSENGER Neutron Spectrometer (NS) Experiment Data Records (EDRs) to Calibrated Data Records (CDRs) and to Derived Data Records (DDR). The processing steps described in this document represent the state of knowledge at the date of this document.

2 Introduction

The Engineering and Science EDRs are the raw data records used to derive NS data used for scientific and engineering analysis. The Engineering and Science EDRs contain the Mission Elapsed Time (MET) and the data for various data products. The NS EDR data types used are the Full Science Packets (FSP), the Galactic Cosmic Ray (GCR) packets, and the Status (STA) packets, which contain NS engineering data. 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. In addition, various reversible, time-series corrections are made to the raw, science data to enable science analysis. Before the engineering data can be used for engineering analysis, the raw data must be converted to physical units. All of these conversions yield calibrated data that are stored in Calibrated Data Records or CDRs. Additional steps are carried out to convert the calibrated data to derived data, which are here defined as net neutron counts from the lithium glass (LG) and borated plastic (BP) sensors. The processing steps from the EDR to CDR to DDR levels are described in this document and include:

1. Description of the coordinate systems used in the NS EDR-to-CDR reduction, including the nadir-fixed coordinate system, which is specific to the NS data.
2. Description of the SPICE kernels that are used for analysis of NS science data.
3. Application of reversible, time-series corrections to the NS count rate and spectral data.
4. Conversion of engineering data from engineering units to physical units.
5. Description of how the relevant spacecraft spatial, orientation, and velocity quantities are calculated.
6. Description of how the DDR data products are derived from the CDR products.

There are four types of NS CDR data products, which are NS_CDR_SPECTRA, NS_CDR_COUNTS, NS_CDR_GCR_SPECTRA, and NS_CDR_ENG. The NS_CDR_SPECTRA files give corrected NS spectra from the various NS sensors as described in the NS CDR/DDR Software Interface Specification (Lawrence and Ward, 2013). The NS_CDR_COUNTS files give correct count rates that are accumulated in the NS EDR data. The NS_CDR_GCR_SPECTRA files give separate spectra from the NS GCR mode. The NS_CDR_ENG data files provide status/engineering from the NS (mostly from the EDR STA packets), but that are now cast into physical units, where relevant.

There are two types of DDR data products, Neutron Count Rate (NS_DDR_NCR) and Neutron Composition (NS_DDR_NCP). The DDR NCR file type contains net count rates (NCR) as derived from the NS_CDR_SPECTRA records. The DDR NCR files contain six new data values, which are the net neutron count rates and uncertainties from the singles LG (LG1 and LG2) and BP sensors. The LG count rates are measurements of thermal and epithermal neutrons and the BP count rate is a measure of epithermal neutrons. Fast neutron measurements are already contained in the CDR data as the time-correlated counts and spectra. For the sake of completeness, the various geometry values given in the CDR data files are repeated in the DDR NCR files. The DDR NCR files are given in ASCII text format to allow more ease of use on various computer systems.

The NCP data product contains latitude-binned relative neutron count rates taken during the primary MESSENGER mission from 26 March 2011 to 25 February 2012. Specifically, these data are the exact data points given in Figure 2 and Figure 3A from Lawrence et al. (2013). The delivered values include measured and simulated data for fast and epithermal neutron data. The simulated data include two cases: one case assumes there is no hydrogen in Mercury's north polar radar bright regions; the other case assumes that the north polar radar bright regions contain 100 wt.% H₂O.

3 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.

For planetary orbital and flyby data, the NS-specific data have two additional coordinate systems for expressing spacecraft velocity and attitude data: the nadir-fixed coordinate system and the spacecraft-fixed coordinate system.

The nadir-fixed coordinate system is defined as follows:

- The z-axis is defined as the vector from the spacecraft center to the flyby or orbiting planet center.
- The nadir-fixed y-axis is defined as the cross product of the nadir z-axis and the spacecraft velocity vector (where both are expressed in the J2000 coordinate system).
- The nadir fixed x-axis is defined as the cross product of the nadir fixed y-axis with the nadir fixed z-axis.

The spacecraft-fixed coordinate system is defined as follows:

- The spacecraft-fixed x-axis is the vector normal to the NS LG1 sensor and is parallel to the spacecraft solar panel booms.
- The spacecraft-fixed y-axis is in the direction from the spacecraft down the magnetometer boom.
- The spacecraft-fixed z-axis is the viewing direction from the instrument deck.

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

- The beginning time of observation is used for the geometric element computations.
- 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 near-point of the body-to-target-center vector.
- 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.
- The SPICE toolkit for IDL (Interactive Data Language), along with the MESSENGER SPICE kernel files were used for calculating the geometric parameters. In the calculation descriptions given in Section 8, direct lines of IDL code are shown in the `courier` font.

4 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, C, IDL*, and MATLAB* at the NAIF web site (<http://naif.jpl.nasa.gov>). The MESSENGER NS CDR processing routines are written in IDL 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 NS 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 NS instrument kernel contains references to mounting alignment, operating modes, and timing as well as internal and field of view geometry for the MESSENGER NS.
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).

MESSENGER SPICE kernels are archived at the NAIF node.

5 Time-Series Corrections to Count Rate and Spectra Data

Time-series corrections are those corrections made to the data to correct or modify the data to account for various internal or external factors, such various accumulation times, internal deadtime corrections, or variations due to time-varying galactic cosmic rays. Examples of how such corrections are carried out for completed planetary neutron and gamma-ray datasets are shown in Maurice et al. (2004) and Lawrence et al. (2004).

For MESSENGER NS data, the primary time-series correction made for the CDR data is normalization to the accumulation time. During different portions of the mission, the accumulation time, which is commandable, is varied. For example, during the three Mercury flybys, the accumulation time was varied from 50, 20, and 2 seconds during different portions of the flyby. In order to make sure all count or spectral data are expressed in units of counts per second, all count data in the NS_CDR_SPECTRA, NS_CDR_COUNTS, and NS_CDR_GCR_SPECTRA files are normalized to the accumulation time from when they were

collected. The given CDR data are thus the time-averaged count rate for a given time bin. To enable the statistical character of the data to be derived, the accumulation time for each time bin is provided in the CDR data. Since the engineering data provide “snapshot” values of various parameters (e.g., voltages, currents, etc.), a normalization with accumulation time is not done with engineering data.

At this time, no additional time-series corrections are made to the NS count or spectral data. However, as the MESSENGER mission moves into the orbital phase around Mercury, it is expected that additional corrections will be made to the data, such as corrections for varying gain, deadtime, and GCR variations. In order to properly characterize what corrections are needed for the orbital data, orbital data first needs to be collected and analyzed. When this is done, the CDR data, along with this document, will be updated to reflect any changes to the processing.

UPDATE AFTER END OF PRIMARY MISSION: The time-series corrections discussed above have been considered after receiving data from the MESSENGER primary mission. Based on the analysis of the orbital data, it has been decided that the CDR products will not be modified for additional time-series corrections. Corrections due to variable gain in the NS sensors will be carried out for the Derived Data Record (DDR) dataset as described in Section 7. Corrections for deadtime are carried out in the CDR data product because these corrections are small compared to other variations in the data. Corrections for GCR variations are not included in the CDR data product. As a guide it is recommended that variations in the NS triple coincidence counter when far from Mercury (altitude greater than 10,000 km) be used as a proxy for GCR variations.

6 Conversion of Raw Engineering Data to Physical Units

For selected data values that measure physical quantities (e.g., temperature, voltage, current), the raw engineering data from the NS EDR status packets are converted into values associated with those physical units using 3rd order polynomial equations, $val = a_0 + a_1x + a_2x^2 + a_3x^3$, where val is the new value in physical units, the a_i values are the coefficients, and x are the EDR data values. Table 1 lists the engineering data values along with the polynomial coefficients that convert the EDR values into CDR values. The polynomial coefficients were determined using preflight calibration information. Note that this table only includes those engineering data values that require conversion into physical units. Other engineering data values that do not require any conversion (e.g., CMD_EXEC, or the number of commands executed) are directly passed through from the EDR to CDR data.

Data Name	Description	a_0	a_1	a_2	a_3
PLUS_5V	+5 volt monitor in volts.	0.0	2.479×10^{-3}	0.0	0.0
NEG_5V	-5 volt monitor in volts.	0.0	2.479×10^{-3}	0.0	0.0

PLUS_12V	+12 volt monitor in volts.	0.0	2.479×10^{-3}	0.0	0.0
NEG_12V	-12 volt monitor in volts.	0.0	2.479×10^{-3}	0.0	0.0
EXTERNAL_CH1	External channel 1. ADC board temperature in units of degrees Celsius.	119.62	-0.047843	9.0349×10^{-6}	-8.2297×10^{-10}
EXTERNAL_CH2	External channel 2. Pre-amp board temperature in units of degrees Celsius.	106.39	-0.034401	4.55×10^{-6}	-3.4218×10^{-10}
EXTERNAL_CH3	External channel 3.	0.0	1.0	0.0	0.0
EXTERNAL_CH4	External channel 4. NS sensor temperature in units of degrees Celsius.	-292.53	0.4096	0.0	0.0
EXTERNAL_CH5	External channel 5.	0.0	1.0	0.0	0.0
PLUS5V_I	+5V current monitor in amps.	0.0	2.44×10^{-4}	0.0	0.0
NEG5V_I	-5V current monitor in amps.	0.0	2.44×10^{-4}	0.0	0.0
PLUS12V_I	+12V current monitor in amps.	0.0	2.44×10^{-4}	0.0	0.0
NEG12V_I	-12V current monitor in amps.	0.0	2.44×10^{-4}	0.0	0.0
TEMP_MON	Temperature monitor in degrees Celsius.	-51.959	0.020821	-9.5358×10^{-7}	1.7064×10^{-10}
PRIMARY_I	Primary current monitor in amps.	0.0	2.44×10^{-4}	0.0	0.0
SWITCHED_PRI_I	Switched primary current monitor in amps.	0.0	2.44×10^{-4}	0.0	0.0
VOLT_LG_HVPS	Voltage of LG HVPS(1) in volts.	-1.2	0.0318634	0.0	0.0
VOLT_BP_HVPS	Voltage of BP HVPS(2) in volts.	-1.2	0.0318634	0.0	0.0
TEMP_LG	Temperature of LG in degrees Celsius.	-280.2	0.034591	0.0	0.0
TEMP_BP	Temperature of BP in degrees Celsius.	-274.32	0.033976	0.0	0.0
HVPS_SPARE1	Spare column. Any values contained are a consequence of the flight software and should be ignored.	-274.32	0.033976	0.0	0.0
HVPS_SPARE2	Spare column. Any values contained are a consequence of the flight software and should be ignored.	-274.32	0.033976	0.0	0.0
HI_VOLT_SETP_BP	Currently commanded High Voltage set point for BP HVPS.	0.0	0.732422	0.0	0.0
HI_VOLT_SEPT_LG	Currently commanded High	0.0	0.732422	0.0	0.0

	Voltage set point for LG HVPS.				
HVPS_1_MAX_VAL	Maximum commandable HVPS 1 value.	0.0	0.732422	0.0	0.0
HVPS_2_MAX_VAL	Maximum commandable HVPS 2 value.	0.0	0.732422	0.0	0.0
DEAD_TIME	Dead time in units of microseconds/16.	0.0	16.0	0.0	0.0

Table 1. Polynomial parameters used for converting engineering data to physical units.

7 Derivation of Net Neutron Count Rates (NS_DDR_NCR)

There are three main steps used in deriving the net neutron count rates from the CDR singles spectra. These steps are: selection of valid data, gain correction of spectra, background subtraction.

7.1 Data Selection

During the orbital mission, there are times when solar energetic particle (SEP) events hit the MESSENGER spacecraft with a large flux of energetic particles. These energetic particles create a large background in the NS sensors that prevents a valid derivation of net neutron counts. Data taken during times of SEPs are therefore removed prior to the derivation of net neutron count rates. SEPs occur rarely enough that their time locations are identified by hand and these times are put into a text lookup file. The exclusion times for SEP events are given in the attached data confidence notes (see catalog/ddr_ds.cat).

In addition to SEP events, there are a small number of times when the NS high voltage (HV) was turned off. These no-HV times were due to spacecraft maneuvers that require instrument high voltages to be turned off, but instrument power was still turned on. These no-HV times are not used in the DDR analysis.

7.2 Gain Corrections

The NS LG and BP sensors identify neutrons via neutron capture reactions in either the LG or BP scintillators. These reactions create energy peaks in measured pulse height spectra corresponding to the energy deposition for a given reaction (Goldsten et al., 2007). During the orbital mission, various factors (temperature, count rate) cause the gain of the pulse height measurement to vary. Specifically, a variable gain means that the channel location in a pulse height spectrum corresponding to a given energy deposition is not constant. Prior to background subtraction, these variable gain measurements need to be equalized so that all measurements have an equal energy scale at equal channels. The gain variation is characterized in each sensor by measuring the position of known peaks as a function of time. For the LG sensors, this known peak is taken from the GCR spectra, where the peak corresponds to the minimum ionizing

energy deposition of cosmic rays in the LG sensors. GCR data from each LG sensor are accumulated over five minutes and the peak position is measured. A running table of these measured peak positions is stored for correction. For the BP sensor, the known peak is taken from the second pulse of the time-correlated data where a clean, low-background peak from fast neutrons is obtained. These second interaction data are accumulated over ten minutes and the peak position is measured. A running table of these measured peak positions is stored for correction. All peak position data are interpolated to each data collection time increment. With these peak position data, the LG and BP pulse height spectra are corrected to a common gain value using the gain correction algorithm described by Lawrence et al. (2004).

7.3 Background Subtraction

After data selection and gain correction, background subtraction algorithms are used to derive net neutron counts from each measured time step. The LG and BP data use different algorithms.

LG Background Subtraction: The algorithm used to derive the net neutron counts for the LG sensors is illustrated in Figure 1. 20-second data accumulations from each LG sensor are shown in Figure 1 as the solid black line. To better isolate the peak, an empirical background function is subtracted from the main spectra. This background function (dotted, black line) is the averaged, high-altitude spectra for each given day, where high-altitude is defined to include spectra collected when the spacecraft was above 8000 km. A background-subtracted spectrum is calculated for each time step as the difference between the original spectrum and the average background spectrum. The background-subtracted spectrum is fit to a Gaussian plus second-order polynomial function to determine both the peak and remaining background counts. Of the three Gaussian parameters (peak position, peak width, peak height), only the peak height is allowed to vary as the other parameters are constrained by gain-corrected values that are empirically determined using data accumulated over long periods of time. This procedure therefore assumes that the gain-corrected peak parameters remain constant and only the peak height varies. The three parameters of the polynomial portion of the fitting function are allowed to vary. The net neutron count rate is the area under the Gaussian portion of the fitting function. The uncertainties in the net neutron count rates are the calculated uncertainties due to the fitting procedure.

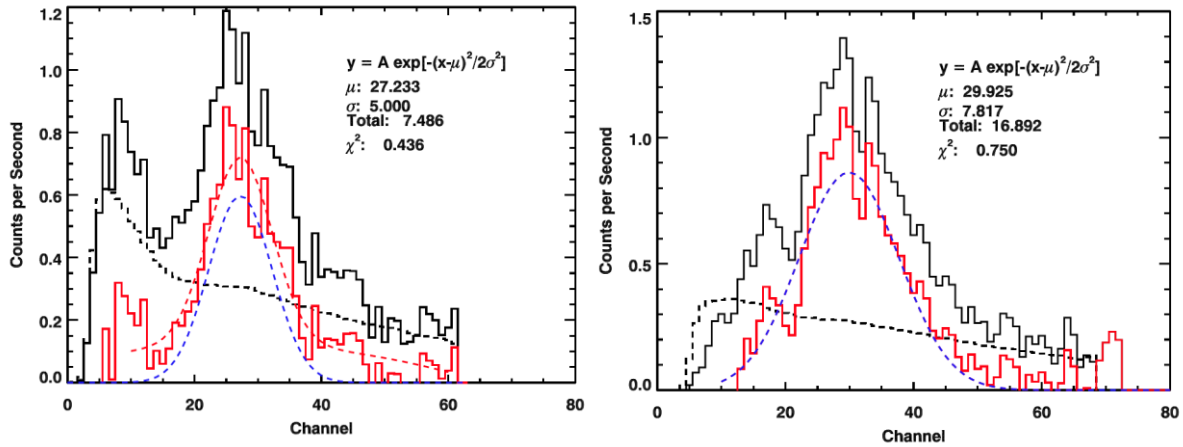


Figure 1. Pulse height data from a single 20-second time collection from the LG1 (left) and LG2 (right) sensors. The black, solid line shows the original, gain corrected data. The dotted black line shows the high-altitude background spectra. The solid red line shows the background-subtracted spectra. The dotted, red line shows the fit of a Gaussian plus polynomial function to the background-subtracted spectra. The dotted, blue line shows the Gaussian portion of the fitted spectra.

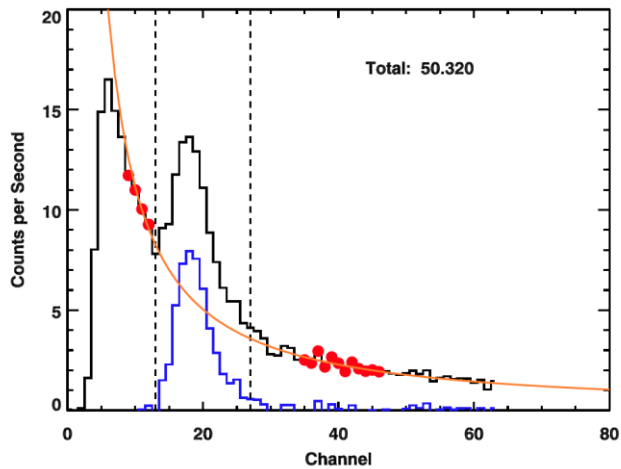


Figure 2. Pulse height data from a single 20-second time collection from the BP sensor. The black, solid line shows the original, gain-corrected data. The red circles show the channel position used for the power law fit. The orange, solid line shows the power law fit. The blue, solid line shows the background-subtracted spectra. The vertical dashed lines show the channel range for deriving net count rates.

counts within fixed channel limits (vertical dashed lines). The uncertainty is the propagated statistical uncertainty for the counts within the channel range.

This fitting procedure is being used because the background LG spectra do not show a clear, simple, functional form, as do the BP data. As more analysis work is carried out for the LG data, it is possible that a revised algorithm may be used.

BP Background Subtraction: The net neutron counts from the BP sensor are determined differently than for the LG sensors. For the BP sensor, a power-law background function is calculated for each time increment of data. Counts from channels both below and above the neutron peak are used to determine the power law. Figure 2 shows a 20-second BP spectrum where this power-law background is determined. This power-law background is then subtracted from the main spectrum to obtain a residual spectrum that dominantly contains the net neutron counts. The net counts are the summed

8 Analysis of Net Neutron Count Rates (NS_DDR_NCP)

NS data analysis has been carried out with empirically derived corrections applied in parallel with a neutron count rate simulation (Lawrence et al., 2013 including supplementary material). The count rate simulation, which was validated with flyby data (Lawrence et al., 2010), accounts for the near-surface production of neutrons by galactic cosmic rays, their transport to the spacecraft, and their detection by the NS. The simulation was used to guide and constrain the empirical corrections and to provide a capability for bounding the surface hydrogen concentrations. The NS analysis requires corrections to account for nonisotropic solid angle variations, spacecraft obscuration effects, time variations in the incident cosmic ray flux, near-surface temperature variations, and variations from a radial velocity Doppler effect. The radial Doppler effect arises because the speed of the MESSENGER spacecraft in the direction of the spacecraft–planet-center vector has a magnitude (0 to 2 km/s) that is similar to the speed of thermal and low-energy epithermal neutrons (~ 2 km/s). Doppler-induced effects are negligible for fast neutrons but have a magnitude of a few percent for low-energy epithermal neutrons and therefore need to be considered (Feldman et al., 1986).

9 Spatial and Velocity Data Calculation

For the NS_CDR_SPECTRA, NS_CDR_COUNTS, and NS_CDR_GCR_SPECTRA data files, the following spatial data are calculated:

SC_ALTITUDE
SC_LATITUDE
SC_LONGITUDE

For the NS_CDR_SPECTRA data files, the following rotational and velocity values are calculated:

VEL_VECTOR
VEL_NORM
V_DOT_X
SC_TO_NADIR_ROT

These calculations are performed for the j2000 ephemeris time value of the MET. The conversion from the met to j2000 is handled by the IDL SPICE toolkit using the following code:

```
cspice_scs2e,MESSENGER_SC_NUMBER,string(met),j2000_et
```

where the `MESSENGER_SC_NUMBER = -236` is the spacecraft ID, and `j2000_et` is the j2000 ephemeris time value.

9.1 SC_ALTITUDE Calculation

The sub-spacecraft point of MESSENGER on the surface of Mercury is found using the “Near-point” method and finds the spacecraft altitude, `SCALT`. This is handled by the SPICE toolkit using the following code:

```
cspice_subpt, 'near
point', 'Mercury', j2000_et, 'LT+S', 'MESSENGER', spoint, sc_altitude
```

The returned value `spoint` is an array with the sub-spacecraft point on the surface of Mercury at time of `j2000_et` in the Mercury reference frame, and the returned value `sc_altitude` is a reference to the altitude of MESSENGER at the time of j2000.

9.2 SC_LATITUDE and SC_LONGITUDE Calculation

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

```
cspice_reclat, spoint, radius, sc_longitude, sc_latitude
```

where `spoint` is the array with the point on the surface of Mercury returned in the `cspice_subpt` call from the `SC_ALTITUDE` calculation above.

The returned value `mercury_radii` is an array with the radii of Mercury, the returned value `SC_LONGITUDE` is a reference to the longitude of Mercury at the point on the surface of Mercury under MESSENGER and the returned value `SC_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 π . For cruise data, the latitude, longitude, and altitude values are calculated, but are of little use for analyzing NS cruise data, since the spacecraft is far from Mercury. For Venus flyby data, the altitude, latitude, and longitude data are given in reference to Venus.

9.3 VEL_VECTOR_J2000 and VEL_NORM Calculations

The appropriate SPICE CD kernel file is read using the `cspice_furnsh` subroutine. The vector velocity of the MESSENGER spacecraft with respect to Mercury (or Venus for Venus flyby data) in the J2000 reference frame is a temporary data value, and is found using:

```
cspice_spkezr, 'MESSENGER', j2000_et, 'J2000', 'LT+S', 'Mercury', posvel, ltime
```

where `posvel` is a 6-element, time-dependent vector. The last three elements contain the three vector velocity components in the J2000 reference frame:

```
vel_vector_j2000 = posvel[3:*,*]
```

A description of how this J2000 velocity vector is transformed into the nadir-fixed coordinate system is described below in Section 8.5. The scalar velocity `vel_norm` is calculated as a loop over the number of elements in `j2000_et`:

```
vel_norm = fltarr(n_elements(j2000_et))
for i=0,n_elements(j2000_et)-1 do $
    vel_norm[i] = cspice_vnorm(vel[*,i])
```

9.4 v_DOT_X Calculation

`v_dot_x` is the dot product of the spacecraft vector velocity with the spacecraft x-axis direction in the spacecraft fixed reference frame. For the purposes of the NS calculation, the spacecraft x-axis is the normal to NS LG1 sensor, as described in Section 3. To obtain this dot product, the spacecraft unit vector directions and the spacecraft vector velocity need to be in the same reference frame, which will be the J2000 frame. First, unit vectors for the spacecraft fixed frame are defined:

```
xaxis_sc = double([1,0,0])
yaxis_sc = double([0,1,0])
zaxis_sc = double([0,0,1])
```

Next, a rotation matrix from the MESSENGER fixed reference frame to the J2000 reference frame is calculated:

```
cspice_pxform,'MSGR_SPACECRAFT','J2000',j2000_et,pform
```

where `pform` is the time-dependent transformation matrix. A new set of x, y, and z vectors are defined:

```
xaxis_j2000 = fltarr(3,n_elements(j2000_et))
yaxis_j2000 = fltarr(3,n_elements(j2000_et))
zaxis_j2000 = fltarr(3,n_elements(j2000_et))
```

and within a loop, the rotation matrix is applied to spacecraft unit vectors:

```
for i=0,n_elements(j2000_et)-1 do begin
    cspice_mxv,pform[*,*,i],xaxis_sc,tmp
    xaxis_j2000[*,i] = tmp
```

```

    cspice_mxv,pform[*,*],yaxis_sc,tmp
    yaxis_j2000[*] = tmp

    cspice_mxv,pform[*,*],zaxis_sc,tmp
    zaxis_j2000[*] = tmp
endfor

```

Now, both the vector velocity and spacecraft unit vectors are in the J2000 reference frame. The normalized dot product of the vector velocity with the spacecraft x-axis can now be calculated:

```

v_dot_x = fltarr(n_elements(j2000_et))
for i=0,n_elements(j2000_et)-1 do $
    v_dot_x[i] = cspice_vdot(xaxis_j2000[*],vel[*])/cspice_vnorm[*]

```

9.5 VEL_VECTOR Calculation

To calculate the vector velocity in the nadir-fixed coordinate system, the following steps need to be completed: 1) calculate a rotation matrix from Mercury into the J2000 reference frame, so that the spacecraft (x,y,z) position vector (spoint), which was returned in the Mercury reference frame, can be transformed into the J2000 reference frame. 2) Calculate unit vectors in the nadir-fixed reference frame. 3) Transform the vector velocity into the nadir-fixed reference frame by calculating a 9-element transformation matrix using direction cosines between the unit vectors in different coordinate systems. 4) Transform the vector velocity from J2000 reference frame into the nadir-fixed reference using this 9-element transformation matrix.

The Mercury to J2000 rotation matrix is obtained using:

```

cspice_pxform,'IAU_MERCURY','J2000',j2000_et,pform_mer_j2000

```

where `pform_mer_j2000` is the transformation matrix. The planet-to-spacecraft vector in the Mercury reference frame is `spoint` (which was calculated in Section 8.2), and is transformed to the J2000 frame using:

```

r_j2000 = fltarr(3,n_elements(j2000_et))
for i=0,n_elements(fsp)-1 do begin
    cspice_mxv,pform_mer_j2000[*,*],spoint[*],tmp
    r_j2000[*] = tmp
endfor

```

Now the nadir-fixed coordinate system is defined as given in Section 3, where the z-axis is the spacecraft-to-Mercury vector (i.e., the negative of the Mercury-to-spacecraft vector). The other axes are obtained via cross products:

```

nadir_z = -r_j2000
nadir_x = fltarr(3,n_elements(j2000_et))
nadir_y = fltarr(3,n_elements(j2000_et))
for i=0,n_elements(j2000_et)-1 do begin
    ; normalize the z-axis

```

```

nadir_z[*,i] = nadir_z[*,i]/cspice_vnorm(nadir_z[*,i])

; the y-axis is the cross product of the z with the velocity
nadir_y[*,i] = crossp(nadir_z[*,i],vel[*,i])
nadir_y[*,i] = nadir_y[*,i]/cspice_vnorm(nadir_y[*,i])

; the x-axis is the cross product with the y axis
nadir_x[*,i] = crossp(nadir_y[*,i],nadir_z[*,i])
nadir_x[*,i] = nadir_x[*,i]/cspice_vnorm(nadir_x[*,i])
endfor

```

Finally, a 9-element transformation matrix of direction cosines is calculated in the same loop as the final `vec_velocity` value in the nadir-fixed coordinate system (NOTE: since the direction cosines are calculated using a cosine calculation between two vectors, the spacecraft fixed unit vectors, `xaxis_sc`, work just as well as the J2000 x-axis vectors `xaxis_j2000`):

```

vel_vector = fltarr(3,n_elements(j2000_et))
trans2 = fltarr(3,3,n_elements(j2000_et))
for i=0,n_elements(j2000_et)-1 do begin
    xxd = cos(cspice_vsep(xaxis_sc,nadir_x[*,i]))
    xyd = cos(cspice_vsep(xaxis_sc,nadir_y[*,i]))
    xzd = cos(cspice_vsep(xaxis_sc,nadir_z[*,i]))

    yxd = cos(cspice_vsep(yaxis_sc,nadir_x[*,i]))
    yyd = cos(cspice_vsep(yaxis_sc,nadir_y[*,i]))
    yzd = cos(cspice_vsep(yaxis_sc,nadir_z[*,i]))

    zxd = cos(cspice_vsep(zaxis_sc,nadir_x[*,i]))
    zyd = cos(cspice_vsep(zaxis_sc,nadir_y[*,i]))
    zzd = cos(cspice_vsep(zaxis_sc,nadir_z[*,i]))

    trans_tmp = [[xxd,xyd,xzd],[yxd,yyd,yzd],[zxd,zyd,zzd]]
    trans2[*,*,i] = trans_tmp

    vel_vector[*,i] = trans_tmp#vel[*,i]
endfor

```

9.6 SC_TO_NADIR_ROT Calculation

A 9-element transformation matrix between the spacecraft fixed coordinate system and the nadir fixed coordinate system is calculated with direction cosines:

```

sc_to_nadir_rot = fltarr(3,3,n_elements(j2000_et))
for i=0,n_elements(j2000_et)-1 do begin
    xxd = cos(cspice_vsep(nadir_x[*,i],xaxis_j2000[*,i]))
    yxd = cos(cspice_vsep(nadir_y[*,i],xaxis_j2000[*,i]))
    xzd = cos(cspice_vsep(nadir_z[*,i],xaxis_j2000[*,i]))

    xyd = cos(cspice_vsep(nadir_x[*,i],yaxis_j2000[*,i]))
    yyd = cos(cspice_vsep(nadir_y[*,i],yaxis_j2000[*,i]))
    yzd = cos(cspice_vsep(nadir_z[*,i],yaxis_j2000[*,i]))

    xzd = cos(cspice_vsep(nadir_x[*,i],zaxis_j2000[*,i]))

```



```

yzd = cos(cspice_vsep(nadir_y[*],zaxis_j2000[*],i))
zsd = cos(cspice_vsep(nadir_z[*],zaxis_j2000[*],i))

trans_tmp = [[xzd,xyzd,xsd],[yxd,yzd,yzd],[zxd,zsd,zsd]]
trans[*],i] = trans_tmp
endfor

```

10 References

- Feldman, W. C., et al., A Doppler filter technique to measure the hydrogen content of planetary surfaces, *Nucl. Instrum. Methods*, A245, 182, 1986.
- Goldsten, John O. et al., The MESSENGER Gamma-ray and Neutron Spectrometer, *Space Science Reviews*, DOI 10.1007/s11214-007-9262-7, 2007.
- Lawrence, D. J., S. Maurice, and W. C. Feldman, Gamma-ray Measurements from Lunar Prospector: Time-Series Data Reduction for the Gamma-Ray Spectrometer, *Journal of Geophysical Research – Planets*, 109(#E7), 10.1029/2003JE002206, 2004.
- Lawrence, D. J., et al., Identification and measurement of neutron-absorbing elements on Mercury's surface, *Icarus*, 209, 195, 2010.
- Lawrence, D.J., W.C. Feldman, J.O. Goldsten, S. Maurice, P.N. Peplowski, B.J. Anderson, D. Bazell, R.L. McNutt, Jr., L.R. Nittler, T.H. Prettyman, D.J. Rodgers, S.C. Solomon, and S.Z. Weider, Evidence for water ice near Mercury's north pole from MESSENGER Neutron Spectrometer measurements, *Science*, 339, 292-296, 2013.
- Lawrence, David J., and Jennifer G. Ward, MESSENGER Neutron Spectrometer Calibrated and Derived Data Record Software Interface Specification, Version 3.1, May 29, 2013.
http://pds-geosciences.wustl.edu/messenger/mess-e_v_h-grns-3-ns-cdr-v1/messns_2001/document/ns_cdr_ddr_sis.pdf
- Maurice, S., D. J. Lawrence, W. C. Feldman, R.C. Elphic, and O. Gasnault, Reduction of Neutron Data from Lunar Prospector, *Journal of Geophysical Research – Planets*, 109(#E7), 10.1029/2003JE002208, 2004.

11 Revision History

Rev	Date	Author(s)	Description	Sections
1.0	2Feb2010	David Lawrence	Initial version.	All
2.0	30May2012	David Lawrence	Added CDR-to-DDR data processing information.	All
3.0	12Sep2012	Jennifer Ward	Peer review edits.	All
4.0	5Jun2013	Jennifer Ward	Added information for new NCP product.	All
5.0	16Jan2016	Susan Ensor	Final mission edits.	All

6.0	1Feb2016	Jennifer Ward	Minor typo fixes and formatting.	All
7.0	9Feb2016	Jennifer Ward	Minor typo fix.	1