The Effect of Faults on Plasma Particle Detector Data Reduction

Michael L. Rilee, Scott A. Boardsen, and Maharaj K. Bhat
Emergent ITI, NASA Goddard Space Flight Center, Mailstop 931, Greenbelt, MD 20770
Phone: (301) 286-4743, email: Michael.L.Rilee.1@gsfc.nasa.gov

Steven A. Curtis
NASA Goddard Space Flight Center, Code 695, Greenbelt, MD 20770
Phone: (301) 286-9188, email: Steven.A.Curtis.1@gsfc.nasa.gov

Abstract—Missions in NASA's Solar-Terrestrial Probe line feature challenges such as multiple spacecraft and high data production rates. An important class of scientific instruments that have for years strained against limits on communications are the particle detectors used to measure space plasma density, temperature, and flow. The Plasma Moment Application (PMA) software is being developed for the NASA Remote Exploration & Experimentation (REE) Program's series of Flight Processor Testbeds. REE seeks to enable instrument science teams to move data analyses such as PMA on board the spacecraft thereby reducing communication downlink requirements. Here we describe the PMA for the first time and examine its behavior under single bit faults in its static state. We find that $\sim$90% of the faults lead to tolerable behavior, while the remainder cause either program failure or nonsensical results. These results help guide the development of fault tolerant, non-hardened flight/science processors.

TABLE OF CONTENTS

1. INTRODUCTION
2. PLASMA MOMENT APPLICATION
3. TESTING METHOD
4. RESULTS
5. CONCLUSIONS

1. INTRODUCTION

FUTURE missions in NASA's Solar-Terrestrial Probe (STP) line have challenging requirements involving multi spacecraft operations and science data communication. These missions form an important part of NASA Sun-Earth Connections theme [1,2]. One such mission is STP's Magnetospheric Multi Scale (MMS, launch 2007), which consists of five spacecraft each carrying a full complement of science instruments designed to observe the fundamental processes that structure Geospace, the near Earth space environment [3]. To fit within mission resource budgets, MMS science return is purposefully limited by obtaining data at a high rate during "burst mode" operations during limited portions of MMS orbits. For the rest of the orbit, "normal mode" stores low resolution data for later transmission to Earth. By performing some data analyses on board the spacecraft, the science return of missions such as MMS may be dramatically enhanced [4].

U.S. Government work not protected by U.S. copyright. Published in the Proc. of the 2001 IEEE Aerospace Conference in Big Sky, MT.

For example, the particle detectors used in space physics experiments measure properties, e.g. mass, momentum, and energy, of the particles making up the space environment. For many years these science instruments have been capable of producing more information than can reasonably be downlinked to Earth. Thus communication limitations have restricted the quality of the raw measurements which in turn limit the conclusions that we may draw from our observations. The STP component of NASA's Remote Exploration and Experimentation Project (REE) is developing methods of on board data analysis that will enable STP missions that are currently infeasible due to communication limitations [5].

Before science data can be used to achieve mission goals, important calibrations and reductions must be performed. Results of these analyses require storage that is often thousands of times smaller than the storage required of the original raw data. By performing these analyses on board spacecraft and transmitting the results to Earth one gains the opportunity to reduce science communication requirements by thousands as well. However, performing these analyses in real-time on board spacecraft requires computing capability that is not feasible if obtained by traditional techniques of constructing radiation tolerant systems. The REE Testbed is a prototype flight processor that uses a mixture of hardware and software techniques to mitigate the effects of radiation induced faults while providing supercomputing performance.

In this work, REE/STP presents a software application being developed for the REE Testbed that analyzes particle detector data. The Plasma Moment Application (PMA) calculates plasma (fluid) moments including the space plasma density, bulk flow, pressure, and heat flux. These moments are calculated using spatial discretization techniques reminiscent of finite-element analysis (FEA). These unstructured grid techniques allow the analysis of almost arbitrarily constructed or edited data sets. Model fitting and parameter estimation will be used to address inescapable "blind spots" and other imperfections in particle detector data. PMA represents a first step towards autonomous space science data analysis, and this paper is, to our knowledge, the first description of how a data analysis code developed within the space science community responds to faults.

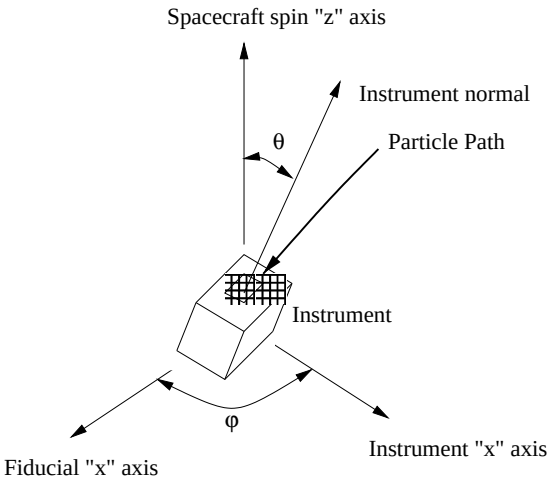

Figure 1. Particle detector geometry

2. Plasma Moment Application

The Plasma Moment Application consists of a model particle spectrometer and an analysis program. The model particle spectrometer produces a set of sensor readings and related information; from these sensor readings, sensor configuration, and spacecraft attitude, a map of the plasma in velocity space is constructed. The analysis program converts the sensor readings to a plasma particle density per unit volume of phase space (velocity $\times$ position). After this conversion, the analysis program fits an anisotropic temperature plasma distribution to the data; this fit is used to patch data gaps and as a check on subsequent analysis. The analysis program then constructs a grid from the distribution of sensor readings which it then uses to calculate statistical moments by numerical quadrature. Though motivated by existing plasma particle spectrometers, these models are implemented at a slightly more abstract level so that they may be specialized to a variety of detectors and measurement strategies.

PMA Spectrometer Model

Our initial spectrometer model is loosely based on the Electron Spectrometer Experiment (ESE) on ISEE-1 [6]. This device is designed to observe the three dimensional electron plasma distribution function as the spacecraft moves through Geospace including the solar wind, magnetosheath, outer magnetopause, and near tail regions. At the output of each of six electrostatic analyzers a two channel electron multiplier registers the influx of electrons from given differential energy ranges and directions. The ESE logarithmically steps through 16 energy levels, dwelling equally at each, and completes one energy sweep every half second. Each electron spectrometer is paired with another, and the pairs are arranged on three mutually orthogonal axes. A single detector is depicted in Figure 1. As the ISEE-1 spacecraft rotates every three seconds, the views of the spectrometers are swept across the sky and a three dimensional picture of the electron and their flow in the vicinity of the spacecraft is recorded every half second.

The PMA Spectrometer Model is not constrained to specifically represent the ISEE-1/ESE. Many of the parameters that determine the data production rate of the ESE can be changed so that the requirements of other particle observation strategies can be studied. These parameters include: the number of energy steps, the central energy and range of each step, the step rate, and the number of detectors. The effective aperture of the instrument and the solid angle from which particles are admitted can also be specified. At the moment we presume ideal efficiency, in that each entering particle is counted. The spacecraft's, and hence the instrument's, rate of rotation may also be specified. These parameters control how particle velocity space is sampled.

To determine the flux of particles into the instrument, a small set of simple plasma models has been implemented. These models map the particle velocity space to a real valued phase space distribution function. The model we use here is a simple Gaussian, or normal distribution, centered on the mean drift of the plasma relative to the spacecraft, and with a width determined by the plasma temperature. The integration of this particle distribution over a sample region of velocity space yields the average number of particles that enter the detector, neglecting statistical fluctuation. To model the actual number of particles entering the detector we select a random number from a Poisson distribution with the aforementioned average as its mean. This process is the inverse of the data reduction which is described in more detail below.

The preceeding two paragraphs have dealt with how velocity space is sampled by a model instrument and how the contents of velocity space are determined. Thus as the instrument proceeds through its observations it builds up a data set of measurements and relevant auxillary data including time and duration of observation, and instrument configuration and position. These data and metadata are read into and processed by the analysis portion of the PMA application.

PMA Analysis Program

The instrument-modelling portion of the PMA reads a file that contains parameters which define a model spectrometer and plasma and produces a set of files that contain flux measurements and descriptive information. The analysis portion of the PMA reads these files with the objective of calculating the statistical moments of the observed plasma distribution in velocity space. The moments to be found correspond to the plasma density, bulk velocity, temperature, and heat flux. The steps required to transform the measurements into the desired fluid moments are outlined in Table 1.

Portions of the PMA program are not exercised for this work. The calibration step mentioned in Table 1 is usually iterative in nature, and not qualitatively different from the "patching step". The patching step is similarly truncated for the runs presented here. The model used for patching the data is constructed to exercise the linear algebra library; the model thus obtained was saved for later comparison to the results of the moment calculation. However, for this work the actual patch is not constructed.

TABLE 1
OUTLINE OF PMA PROGRAM

1. Input data
2. Transform to working coordinates & units
3. Calibrate
4. Patch holes in data set
5. Construct grid from data
6. Calculate moments
7. Write results

Constructing a patch requires a matrix-vector multiplication and the addition of a "few" modeled data points to the data set. Our experiences with this process will be documented elsewhere, but here the focus is the behavior of PMA under faults, and we feel that an additional matrix-vector multiply will not significantly affect our results.

Here, calibration means accounting for the different sensitivities of the various detectors. These different sensitivities are due to intrinsic variability in these devices as shipped, or differences in how they wear. These differences can be ameliorated by comparing how different detectors respond to a predictable environment, for example, by using the solar wind as a calibration beam. For these instruments the calibration, though important for the analysis of real data, looks like a filter, a model fit, or iterates of these. Because a model fit is required for the patching step, the addition of the calibration step will not add to the present study but will be addressed in future work. Therefore, our models presume ideal (cross) calibration.

Spacecraft often acquire an electric charge, which leads to a spacecraft electric potential $\phi_{\rm sc}$. Charged particles which enter the detector move through this potential, changing their energies. Therefore, in reconstructing the PDF from the sensor data, we must account for the shift ΔE in particle energy E:

$$\Delta E = q\,\phi_{\rm sc}. \tag{1}$$

This energy shift is usually small and the velocities low, so nonrelativistic formulae may be used. Current code simply drops data with $E < q\phi_{\rm sc}$, and fits the patch to higher energy, $E > q\phi_{\rm sc}$, data.

Model Fitting

Model fitting is one method that has been used to address irregularities in plasma analyzer data. Particles with low velocities relative to the spacecraft are particularly difficult to measure for two reasons. First, a lower velocity implies a lower flux into the instrument which implies lower count rates and greater uncertainties and fluctuations in particle flux. Second, lower velocity particles are more susceptible to the electric potential of the spacecraft. A spacecraft placed in the Sun's ultraviolet radiation becomes electrically charged and drives an electric current in the surrounding plasma that "corrupts" measurements of the properties of the space plasma. For these two reasons, measurements of the plasma particle distribution function at lower energies are difficult to interpret and often look like a "hole" in the data. A way around this problem is to "patch" the hole by fitting a model distribution function, e.g. a Maxwellian or Gaussian, to the measurements and then using this model to generate model data in the hole.

In this work, we use a ten parameter generalized Maxwellian (Eq. 2). The parameters are of the following form:

$$\ln f \approx B_0 + \mathbf{B}\cdot\mathbf{v} + \mathbf{v}^\top\cdot\mathcal{B}\cdot\mathbf{v}, \tag{2}$$

where $\mathcal{B}$ is an upper triangular 3×3 matrix; the symbol $^\top$ denotes transposition. Note that $\ln f$ is a column vector of data, in the computer scientific sense, indexed by $\mathbf{v}$. Hence the model $\ln f$ is linear in the parameters $B = (B_0, \mathbf{B}, \mathcal{B})$, Eq. 2 can be rewritten:

$$\ln f = \mathcal{V}\,B, \tag{3}$$

where $\mathcal{V}$ is defined to make Eqs. 2 and 3 equivalent. Furthermore, $\mathcal{V}$ depends on the velocities $\mathbf{v}$ at which the data was obtained. We can write the $j^{\rm th}$ row of $\mathcal{V}$

$$\ln f_j = \left(1,\ \mathbf{v}_j,\ \mathbf{v}_j^\top..\mathbf{v}_j\right)\ B, \tag{4}$$

where j indexes the measurements $f_j, \mathbf{v}_j$. One can test the patch to f by checking for kinks or non-monotonicity.

Because the logarithm of the model distribution function is linear, we use a linear least squares fitting procedure implemented with PLAPACK [7] and LAPACK [8]. PLAPACK (Parallel Linear Algebra Package) is an object oriented library that provides dense linear algebra for multiple processor computers using the Message Passing Interface (MPI) [9]. Furthermore, experiments in Software Implemented Fault Tolerance (SIFT) are being performed with PLAPACK, with the plan that applications such as PMA will gain enhanced robustness to faults simply by using a SIFT based version of PLAPACK. After the model parameters have been found, they can be used to patch irregularities as mentioned above, or they can be compared with the results of the plasma moment calculation to be discussed below.

Plasma Moments

The primary function of the PMA is the plasma moment calculation, which proceeds after calibration and patching have resulted in a consistent, complete data set. To allow a great variety of measurement strategies to be studied, as well as to allow easy import of existing particle spectrometry data, the PMA uses a very general method to partition velocity space using techniques from finite element analysis [10].

The fundamental quantities of interest are the plasma or fluid moments of the distribution of particles in the space environment. The density, bulk flow velocity, and the pressure tensor are

$$n = \int d^3v\, f(\mathbf{v}), \tag{5}$$

TABLE 2

QUANTITIES ASSOCIATED WITH PDF ESTIMATION.

Symbol	Interpretation	Dependencies
t	Time	
ϕ, θ	Azimuth and co-altitude	t
$\mathbf{n}$	Detector unit normal	θ, ϕ
$\mathbf{v}$	Particle velocity	θ, ϕ, t
v	Velocity magnitude	$v = \lvert\mathbf{v}\rvert = -\mathbf{n} \cdot \mathbf{v}$
C	Measured particle counts	$\mathbf{v}$
f	Distribution function	$C, \mathbf{v}, \theta, \phi$

TABLE 3

THE GEOMETRIC FACTOR η CONNECTING THE PDF AND COUNTS.

Symbol	Interpretation	Dependencies
A	Detector collecting area	t
$\delta\Omega$	Solid angle subtended	$\delta\Omega = \sin\theta\,\delta\theta\,\delta\phi$, E
δt	Measurement duration	t
$E^{-1}\delta E$	Energy bandpass	t
η	Integration factor	$\eta = A\delta\Omega\delta t\,(2\,E)^{-1}\delta E$

$$\mathbf{u} = \frac{1}{n}\int d^3v\, f(\mathbf{v})\,\mathbf{v}, \text{ and} \tag{6}$$

$$\mathcal{P} = m\int d^3v\, f(\mathbf{v})\,(\mathbf{v}-\mathbf{u})(\mathbf{v}-\mathbf{u}), \tag{7}$$

where we interpret the dyadic product in Einstein's notation thusly: $\mathbf{ab} = a_i b_j$. The integrals are evaluated over the particle velocity space, and m is the mass of the particle, e.g. electron, proton, ionized Oxygen, being studied. The tensor product is defined by: $\mathcal{P}\cdot\mathbf{v} \equiv \mathcal{P}_{ij}v_j$.

The heat flux tensor is given by:

$$\mathcal{Q} = \frac{m}{2}\int d^3v\, f(\mathbf{v})\,v^2\mathbf{v} - \left[\frac{1}{2}\mathbf{u}\,\mathrm{Tr}\mathcal{P} + \mathcal{P}\cdot\mathbf{u} + \frac{m\,n}{2}\mathbf{u}\,u^2\right], \tag{8}$$

where $u = \lvert\mathbf{u}\rvert$. Finally, in practice it is not always necessary to evaluate n, $\mathbf{u}$, $\mathcal{P}$, $\mathcal{Q}$, as written above. For some purposes, integrals of the dyadic powers of $\mathbf{v}$ are adequate, and from these the pressure and heat flux can be calculated.

The differential number of particles entering the detector, which ideally equals the number of counts registered by the detector, during a given data gathering interval is

$$C(\mathbf{v}) = -d^3\mathbf{v}\, f(\mathbf{v})\,A\,\mathbf{n}\cdot\mathbf{v}\,\delta t. \tag{9}$$

In the case of nonrelativistic energies, the velocity volume differential may be related to particle detector parameters via Tables 2 and 3 to find:

$$f(\mathbf{v}) \approx \frac{C(\mathbf{v})}{\eta\, v^4}. \tag{10}$$

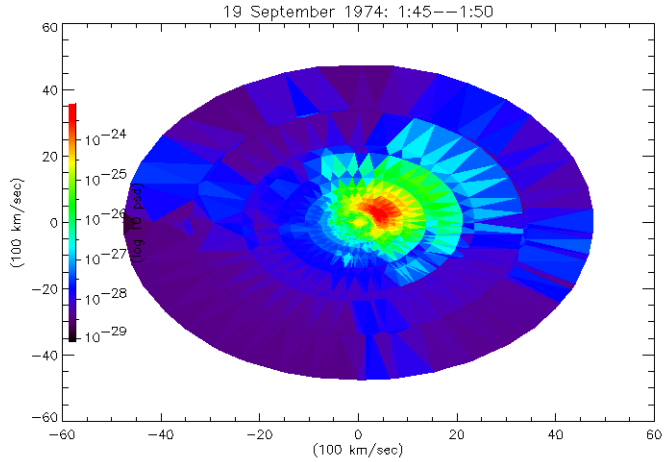

Figure 2. Particle distribution function in PMA data format. Data is from the HAWKEYE spacecraft [11,12].

Unstructured Grid & Quadrature

A three dimensional grid or mesh is constructed in velocity space from the positions of the spectrometer data. The grid is constructed using the Delaunay triangulation implemented in the GEOMPACK library [13,14]. The resulting grid is a set of tetrahedral volumes that partition velocity space; the vertices of these tetrahedra correspond to velocity space measurements. From there it is a simple matter to evaluate functions defined on the measurements, including the calculation of moments of the plasma particle distribution by numerical quadrature. Integrations over velocity space become summations over the set of tetrahedra.

An advantage of this approach is that nearly arbitrary distributions of measurements may be handled. A key disadvantage is one that also arises in FEA, namely the errors that can be introduced by poorly shaped tetrahedra. Poorly shaped tetrahedra lead to difficulties both in grid construction and numerical errors during calculation. Nevertheless, we feel that this FEA inspired approach allows one to extract information from the data set with minimal transformation because the data is left "in place", and not accumulated into bins or scattered onto a grid. Thus our approach will provide the best time resolution possible for a given data set because there is no grid to be filled with data: the data defines the grid. Furthermore, our approach allows particle spectrometer developers to start to consider reactive, adaptive measurement strategies that allow the instrument to focus on features within velocity space, obtaining higher time and velocity space resolution.

Another issue involving grid construction, is that GEOMPACK forms a convex-hull of the data points. The convex hull is the minimal convex surface that encloses all of the points. For samplings of velocity space that are essentially non-convex, this can lead to grids (discretizations) that have copious, ill-shaped tetrahedra as mentioned above. The way around this problem that we have pursued is based on the recognition that in spherical coordinates the regions of velocity space samples are essentially

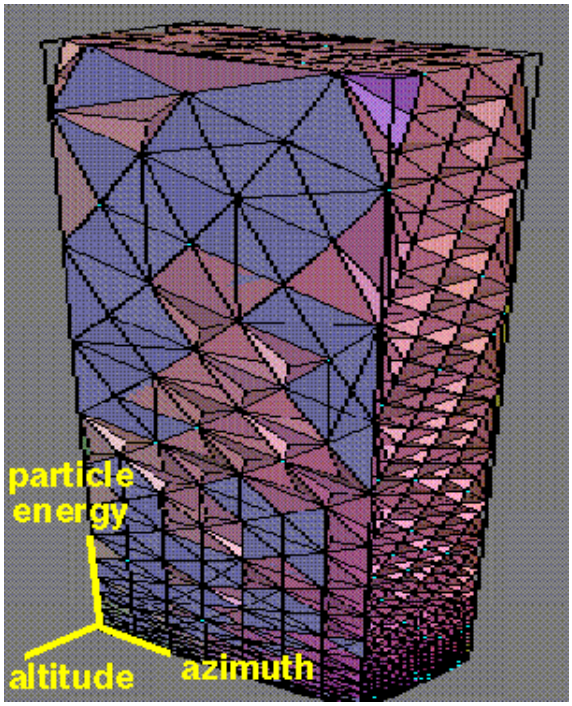

Figure 3. The discretization of velocity space.

convex (Figure 3).

The location of the data points in the angular dimensions are trivially bound, while the magnitude of the velocity has given upper and lower limits. Therefore, the data points lie within a rectangular volume in spherical coordinates. As long as there is a sufficient density of points within this volume, one obtains good results for both the automated discretization and the numerical calculation. Futher details regarding the automated discretization are beyond the scope of this work and will be presented elsewhere, including a discussion of issues relating to troublesome degenerate points.

Let ξ_i denote the direction and velocity state of the detector during measurement i: $\xi_i = \{v_i, \theta_i, \phi_i\}$. With this definition we write the estimate distribution function at ξ_i as f_i. We can then evaluate the integrals Eq. 5–8 by numerical quadrature because we know the integrands as functions of the data: ξ_i and f_i. After the triangulation, a list of tetrahedra is constructed. The volume of each tetrahedra is determined by its four vertices determined by ξ_i. We determine the tetrahedra's contribution to the integrals by linearly interpolating the appropriate functions of f_i and ξ_i from the nodes.

3. Testing Method

Application Form

In this paper we consider the effect of faults on this data analysis scheme. A goal of the REE effort is to allow instrument operators or scientists to run data analysis programs on board the spacecraft. Furthermore, REE seeks to provide a computing environment, based on commercial-off-the-shelf (COTS) components, augmented by Hardware implemented fault tolerance in conjuction with SIFT (H/SIFT), that otherwise closely resembles the computing environment used by the ground-based scientist. Scientist generated analysis programs are to run on the REE Flight Processor and will suffer faults. Therefore, most of the PMA code was constructed using Fortran 90 and 95 with enough C and shell scripting to allow for library access and run management (Table 4). Much arithmetic was performed using double precision floating point.

TABLE 4
Application Environment

Apple PowerBook 2000
384 MB RAM
Yellow Dog Linux Champion Server v1.2
Kernel version 2.2.15pre19
PLAPACK 2.08
GEOMPACK version 3
LAPACK version 2.0 (Very few routines used)
Message Passing Interface MPICH 1.2.0 & p4 [15,9,16]
ABSOFT Fortran 90 version 3.0 & bundled IMSL
GNU gcc 2.95.2
Optimization enabled for C & Fortran compilers

To start to determine how faults affect such science programs, we have subjected the PMA to faults and have studied their effects. REE is in the process of bringing a hardware and software based fault injector on line; this fault injector will allow faults to be injected into various parts of the REE Flight Processor Testbed, including memory, cache, registers, etc. Currently, however, we do not have ready enough access to the run-time state of the PMA and processor to allow extensive testing of the dynamic portions of the PMA process.

On the other hand, we have full access to the static state of the program.[1]. Therefore to start our testing we have pursued the following simple procedure. The PMA analysis executable and model data were archived without compression. This archive was copied and then a randomly chosen bit in the copy was flipped. Every bit had an equal chance to be chosen. The PMA and data were then extracted from the archive and the analysis was run. The run-time environment for the tests used here is detailed in Table 4. The size of the executable and data set are given in Table 5; this shows the relative sizes of the two parts of the application, which implies that the executable likely suffered about 50% more faults than the data. No special effort was expended to reduce executable size or enhance run-time robustness, and the data set used did not model any particular instrument.

[1] By *static state* we mean that state available before run-time. *Dynamic state* is complementarily defined.

TABLE 5

SIZES OF STATIC PORTION OF PMA

PMA Portion	MB: Megabytes KB: Kilobytes.
Data set	1.1 MB
Executable	1.6 MB
Run-control input files	1.5 KB

TABLE 6

RESULTS FROM ERROR TESTING.

Percentage	Number	Characteristic
100.0	1161	Iterations
7.6	88	Severe control flow errors
92.5	1074	Produced n & $\mathbf{u}$
91.0	1057	n within 0.5% of correct
80.7	937	n negligible difference
1.5	16	n intolerably corrupted results
90.4	1050	$\mathbf{u}$ within 0.5% of correct
77.0	894	$\mathbf{u}$ negligible difference
1.7	20	$\mathbf{u}$ intolerably corrupted results

4. RESULTS

The results of a series of runs is tabulated in Table 6. The zeroth moment (density) was examined for discrepancies from the non-corrupted control run. Two tenths of the runs either did not successfully complete, or produced a density that differed from true. However, half of the non-negligibly corrupted results, i.e. one tenth of the runs, produced densities that were within 0.5% of true and are therefore usable. Similar results are obtained if the first velocity moment, i.e. the bulk flow velocity $\mathbf{u}$, is studied.

The types of errors are summarized in Table 7. Of the 88 errors that halted the program, just under half seem to have caught within the "lower" levels of MPICH and its p4 substrate. Another 15% were caught by the operating system (OS), which never faltered during these tests. Just under 25% of the serious errors occured in PLAPACK or higher levels of MPI, e.g. incorrect PLAPACK arguments or faulty MPI communicattion parameters, etc. A number of Fortran run-time errors also occured, though about half of them seem due to corruption of the archive and thus are not directly relevant to the current problem.

Table 8 shows the source of faults that caused the 88 serious errors. Over three fourths of the serious errors were caused by bits flipped in the executable. Package level errors, e.g. PLAPACK/DLINRG, are due mainly to data corruption. Error induced "user aborts" are evenly split between data and executable corruption. For segmentation violations caught in MPI, faults in the executable outnumber faults in the data by nine to one. The faults that produced the 16 intolerable results recorded in the Table 6 did not cause the PMA to crash; consistent with this is that 15 out of those 16 faults occured in data.

TABLE 7

CLASSIFICAITON OF THE 88 SEVERE ERRORS

Number	Type of severe error
42	MPI or p4 caught SIG BUS/SEGV
21	PLAPACK or higher level MPI/user abort
13	OS caught illegal instruction/segmentation faults
10	FORTRAN run-time errors
2	Terminated due to excessive run-time

TABLE 8

LOCATION OF FAULTS LEADING TO THE 88 SEVERE ERRORS

Number	Application area of fault
67	executable
15	data
6	archive (file) structure

5. CONCLUSIONS

In this work we have shown that a data analysis program written by a scientist for a COTS parallel computer and development system can have reasonable performance in the presence of faults to its static state. This is partially due to the robustness of double precision floating point arithmetic to single bit errors. Another reason for the application's robustness is that PMA reduces the data to statistical moments, which reduces the overall impact of data errors. Furthermore, PMA operates as a filter that undergoes only a limited amount of iteration. Therefore, there is not a great deal of time for faults and their consequences to accumulate. Fresh, new processes will continually start and limit the effect of faults to old, expired processes.

Applications that require long lifetimes, servers, operating systems, system monitors, mission planner/schedulers may drive fault tolerance development, but science analyses seem from the start more robust. However, for science analyses that may affect mission operations, stringent real-time requirements on system or instrument health and safety may drive higher levels of science application robustness than we have observed with our experiments with the PMA. Certainly, given the inherent robustness of PMA, achieving these higher levels of reliability seems feasible within the REE technology development framework.

The logical continuation of this work is to perform a similar series of tests using the fault injection capabilities of the REE Testbed. With that system we will be able to determine how PMA responds to errors in its dynamic state. We doubt that the results will differ greatly from what we have found here, though there may be a somewhat wider range of consequences due to the broader range of faults. Certainly, we will gain a better appreciation for the behavior of the PMA when other parts of the computing subsystem suffer faults, because that is something we are currently not able to test. Our results point out that the REE emphasis on hardware, software libraries, and OS fault tolerance is well placed. PMA will be used to test the fault tolerant versions of these COTS components for space borne

applications.

6. Acknowledgements

Acknowledged is support from the Remote Exploration and Experimentation Project of NASA's High Performance Computing and Communications Program which is funded by the Office of Space Science with the work performed under contract NAS5-32350. A. Figueroa-Viñas played an important role during the formulation of this work.

References

[1] NASA, "Solar terrestrial probes program," http://stprobes.gsfc.nasa.gov.
[2] NASA, "Sun-earth connections theme," http://sec.gsfc.nasa.gov.
[3] J. L. Burch, S. A. Curtis, J. Durning, et al., "The magnetospheric multi-scale mission... resolving fundamental processes in space plasmas," Tech. Rep. NASA/TM-2000-209883, NASA Goddard SFC, December 1999.
[4] S. Curtis et al., "Small satellite constellation autonomy via on-board supercomputers and artificial intelligence," in *The 51st International Astronautical Congress*, October 2000.
[5] NASA, "Remote exploration and experimentation program," http://www-ree.jpl.nasa.gov.
[6] K. W. Ogilvie, J. D. Scudder, and H. Doong, "The electron spectrometer on isee-1," *IEEE Transactions on Geoscience Electronics*, vol. GE-16, no. 3, July 1978.
[7] R. A. van de Geijn, *Using PLAPACK: Parallel Linear Algebra Package*, The MIT Press, 1997.
[8] E. Anderson et al., *LAPACK User's Guide*, SIAM, 1995.
[9] W. Gropp et al., *Using MPI: Portable Parallel Programming with the Message Passing Interface*, The MIT Press, Cambridge, MA, 1994.
[10] C. Hirsch, *Numerical Computation of Internal and External Flows*, vol. 1, J. Wiley & Sons, 1988.
[11] R. Kessel et al., "Evidence of high-latitude reconnecting during northward imf: Hawkeye observations," *Geophys. Research Letters*, 1996.
[12] J. Bonnell et al., "On the use of specific entropy for onboard plasma regime characterization," presented at the Spring Meeting of the American Geophysical Union, May-June 2000.
[13] B. Joe, "GEOMPACK – a software package for the generation of meshes using geometric algorithms," *Adv. Eng. Software*, vol. 13, pp. 325–331, 1991.
[14] B. Joe, "Tetrahedral mesh generation in polyhedral regions based on convex polyhedron decompositions," *Intern. J. Num. Meth. Eng.*, vol. 37, pp. 693–713, 1994.
[15] W. Gropp and E. Lusk, "Installation guide for mpich, a portable implementation of mpi," Tech. Rep. ANL-96/5, Argonne National Laboratory, 1996.
[16] R. Butler and E. Lusk, "User's guide to the p4 parallel programming system," Tech. Rep. ANL-92/17, Argonne National Laboratory, IL, 1992.

Michael L. Rilee is a scientist with Raytheon supporting the GSFC Laboratory for Extraterrestrial Physics and the Science Computing Branch of the GSFC Earth and Space Science Computing Division. For NASA's Remote Exploration and Experimentation project he has lead development of the Plasma Moment Application and the Radio Astronomical Imager which are science data analysis applications designed for space borne supercomputers. He is currently researching a High Performance Computing System that may fly on Magnetospheric Multi Scale (launch 2007). At GSFC he has been active in Nano-Satellite technology development and the application of parallel computing to data analysis and astrophysical fluid simulation (PARAMESH). He received his Ph.D. and M.S. in Astrophysics (Plasma Physics) from Cornell University, and his B.A. in Astrophysics and Mathematics from the University of Virginia in Charlottesville, VA.

Scott A. Boardsen is a scientist with Raytheon at GSFC and has fifteen years of experience in Geospace research involving data analysis and modeling and in space mission support activities. He is currently in charge of the daily pointing plans for the imagers on the POLAR spacecraft. He is in charge of developing the simulation model of the radar echoes and plasmagrams for the Radio Plasma Imager on board IMAGE (launched 2000). He plays a key role in developing ISTP key parameter and orbit products and in making them accessible to the Geospace research community. He has a strong background in numerical analysis, computer modeling, and data visualization applied to many areas of endeavor, from data analysis, to instrument modelling, to mission planning. He currently contributes to NASA's Remote Exploration and Experimentation Program.

Maharaj K. Bhat is a computational scientist with Raytheon supporting the GSFC Laboratory for Extraterrestrial Physics and the Science Computing Branch of the GSFC Earth and Space Science Computing Division. He has extensive experience in computational fluid dynamics, parallel algorithm implementation, software development, and experimental techniques. He received his Ph.D. in Aerospace Engineering from the University of Tennessee Space Institute, Tennessee in 1988.

Steven A. Curtis has been the Head of the Planetary Magnetospheres Branch in the Laboratory for Extraterrestrial Physics at Goddard Space Flight Center for the past decade. He is a Principle Investigator in NASA's Space Physics Theory Program (Global Magnetospheric Simulations) and in NASA's Remote Exploration and Experimentation Program (advanced spacecraft onboard computing). He is presently Project Scientist for the 4-5 spacecraft Magnetospheric Multi Scale in which he has played a leading role in conceptualization of the science and spacecraft design implementation. His research interests include the global and mesoscale structure of the magnetosphere and its simulation, atmosphere-magnetosphere interactions, and wave-particle interactions in Geospace. He is particularly interested in the multiscale closure needed between theory and observations. He received his Ph.D., M.S., and B.S. in Physics from the University of Maryland.