INL/EXT-11-23095
Revision 1

BPLU Completion and Verification Report

Dr. George L. Mesina

September 2011

Idaho National Laboratory

The INL is a U.S. Department of Energy National Laboratory
operated by Battelle Energy Alliance

INL/EXT-11-23095

BPLU Completion and Verification Report

Dr. George L. Mesina

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

BPLU Completion and Verification Report

INL/EXT-11-23095

September 2011

Approved by:

George L Mesina — Nov 7, 2011

Dr. George L Mesina — Date
Title [optional]

J.R. Wolf — 11/11/11

Dr. James R. Wolf — Date
Title [optional]

11/9/11

Dr. George W. Griffith — Date
Title [optional]

11/9/11

J. Hope Forsmann — Date
Computer Scientist

ABSTRACT

All User Problems (UP) associated with BPLU since 1999 have been located; this includes the UP listed in Section 1.2, Bullet 1. Input files to test the code against these problems have been obtained, and the code has been debugged. Modifications to the code for these problems have been submitted for inclusion in RELAP5-3D, Version 3.0.2. The modifications have been documented in this report. There are no changes to the RELAP5-3D manual.

CONTENTS

FIGURES

TABLES

ACRONYMS

BPLU	Border Profile LU
BW	bandwidth
FY	Fiscal Year
IRUG	International RELAP5 User Group
PDE	Partial Differential Equation
PVM	Parallel Virtual Machine
TDV	Time Dependent Volume
UP	User Problems

BPLU Completion and Verification Report

1. INTRODUCTION: TASK DESCRIPTION

The Border Profile LU (BPLU) linear equation solver is the default solver for newer versions of RELAP5-3D. The purpose of this task was to make corrections that will preclude future failures. The work was performed on RELAP5-3D, Version 3.0.1 and the updates are incorporated in Version 3.0.2.

1.1 Background

There are over a dozen user problems that have been reported from 1999 until present that involve the BPLU solver in RELAP5-3D. These issues can be combined into two categories of problems with the solver:

- The BPLU solver fails when running multidimensional components with the nearly-implicit hydrodynamics advancement scheme.
- The BPLU solver fails with some input models where the MA18 sparse solver does not fail.

The BPLU solver is the default solver for RELAP5-3D and can significantly reduce execution time compared to the previous default solver, MA18. However, because of the failures noted above, the MA18 solver currently must be used for coupled analyses.

1.2 Requirements

The most recent developmental version of the code must be debugged and tested with a suite of test problems comprised of the following:

1. Problems UP 04021, UP 09024, and UP 09047; these are detailed in Section 2.
2. The current installation test set; this is detailed in Section 6.
3. The Developmental Assessment problems; Reference 1.

The work is complete when the modified code successfully runs these problems. On a Linux operating system, all cases listed above run successfully, including those that require PVM. On a PC, all cases that do not require PVM run successfully.

1.3 Status

All User Problems (UP) associated with BPLU since 1999 have been located; this includes the UP listed in Section 1.2, Bullet 1. Input files to test the code against these problems have been obtained, and the code has been debugged. Modifications to the code for these problems have been submitted for inclusion in RELAP5-3D, Version 3.0.2. The modifications have been documented in this report. There are no changes to the RELAP5-3D manual.

2. LIST OF USER PROBLEMS

This task is to find and correct all outstanding UP and then test the modified coding on all available test cases. All recorded user problems were searched back to Fiscal Year (FY) 1998. The list of UP that relate to BPLU is given in Table 1. The table is grouped by pairs of rows, and the first row of each pair has four columns. In order they are: UP number, STATUS as recorded status prior to the start of the project in June 2011, phrase describing the problem, and name of user who reported it. The second row has two columns: the month and year of the report, and the detailed description and status of the UP.

Table 1. User problems related to BPLU.

<table>
<tr><th>99063</th><th>Resolved Beforehand</th><th>Unphysical Result w/ BPLU</th><th>User Unknown</th></tr>
<tr><td>November 1999</td><td colspan="3">An unphysical result occurred while running on version RELAP5-3D, cft. A nodal kinetics problem (typpwrr.i) was modified by adding boron. When running with the nearly-implicit option (ss = 15 or ss = 31), the boron density was identically constant during the run. If the semi-implicit option is used (ss = 7), the boron density varies (which it should). When the solver was changed from BPLU (Card 1, Option 33) to the default solver, the boron density varies as it should for both the nearly-implicit option and the semi-implicit option.
STATUS
During a FORTRAN 95 conversion of subroutine SIMPLT, it was noted that for the “do-m” loop immediately after the “do-1410” loop, the index “i” for array boron was not calculated unless the pre-compiler flag “dssrtran” was active. It was corrected by making the calculation of “i” unconditional. This correction will be in Version 2.3.7.</td></tr>
<tr><th>04021</th><th>Resolved Beforehand</th><th>Pre-transient Failure in BPLU</th><th>User Unknown</th></tr>
<tr><td>April 2004</td><td colspan="3">In running the AP600 input deck (ap3dsbs.i) on Version 2.3.2+updates, using BPLU and the nearly-implicit scheme, the calculation fails in subroutine BPPART. The error message read: “chain connects to a previous chain, Error: chain, member, neighbor = 11 2833 2601”. The same deck runs with BPLU and the semi-implicit scheme.
STATUS
Solved by giving BPLU its own arrays in F95 version.</td></tr>
<tr><th>04049</th><th>Resolved Beforehand</th><th>PVM Coupling Failure in BPLU</th><th>User Unknown</th></tr>
<tr><td>April 2004</td><td colspan="3">While testing an IRUG code version based on RELAP5-3D 2.3.6, it was found that the semi-implicit PVM coupling problems fail when using the default solver BPLU, but run when using the sparse solver MA18.
STATUS
Solved by giving BPLU its own arrays in F95 version.</td></tr>
<tr><th>04052</th><th>On-hold Beforehand</th><th>Unphysical Result Coupling to CONTAIN</th><th>User Unknown</th></tr>
<tr><td>October 2004</td><td colspan="3">While debugging a set of updates to the heat flux coupling of CONTAIN to RELAP5-3D, an unphysical result was found. The results of the CONTAIN coupling test case are different when using the BPLU and sparse matrix solvers. Further investigation showed that the unmodified version of 2.3.6 has the same problem (i.e., the CONTAIN coupling test case has different results when using the sparse matrix solver than when using the BPLU matrix solver).
STATUS
On hold</td></tr>
</table>

06007	**Solved Beforehand**	**Problem with 3D Component Fails**	**Takashi Nagae, INSS**
January 2006	A MULTID calculation shows incorrect trip times on Version 2.3.6, but correct trip times on Version 2.2.4. **STATUS** An e-mail was sent to Takashi requesting the input deck or a subset of the input deck that demonstrates the trip problem. Takashi sent an e-mail with four input decks. The e-mail explained there are really two separate problems: The first problem is incorrect trip times on Version 2.3.6. The second problem is another issue when the MULTID component is used. It has been tested the decks on Versions 2.2.4 and 2.3.6 on the Dec Alpha. The trip time input decks (flow18.i and flow18_ss.i) are an initial calculation followed by a restart calculation. On Version 2.2.4, the trip times are correct for both the initial calculation and the restart calculation. On Version 2.3.6, the trip times are correct for the initial calculation. They are incorrect (have values of e-154, e-309, etc.) for the restart calculation and the calculation then fails with a floating invalid in subroutine TRIP. The MULTID input decks (test1.i and test2.i) are also an initial calculation followed by a restart calculation. On Version 2.2.4, the code runs for both the initial calculation and the restart calculation. On Version 2.3.6, the initial calculation runs. The restart calculation fails in input processing (for the default solver BPLU); the message is: 'inconsistency in reordering, check dimensions', 'bpform -1'. The code stops in BPLU subroutine BPFORM. Reran the initial and restart calculations with the MA18 solver; the restart calculation now hangs in input processing, and it appears to be in subroutines TSETSL or PMINVD. The first problem was corrected by initializing the variable trptimss in rtrip.F. The update was submitted for Version 2.5.4. A workaround exists for Version 2.3.6 by using the 107 card and enabling trips. The second problem appears to be caused by insufficient memory. This means that an input problem with a large MULTID component will now fail if the user selects the stdy-st option, which selects the nearly implicit scheme by default. A workaround is to use the 107 card to allow the user to select the semi-implicit scheme through controls on the time step cards. This second problem was designated as UP#06013.		
06010	**On-hold Beforehand**	**BPLU Scratch Memory Overwrite**	**Dr. Walter Weaver, INL**
January 2006	The BPLU solver overwrites data in scratch space when executing a large problem coupled to Fluent. **STATUS** The user must use the sparse matrix solver for PVM coupled problems.		
06013	**On-hold Beforehand**	**Problem with 3D Component Fails**	**Takashi Nagae, INSS**
February 2006	While working on UP#06007, it was discovered that problems with large MULTID components would not run with the nearly-implicit solution scheme option (which is now the default in “stdy-st” mode). **STATUS** With the BPLU solver, the code will execute for 6 × 6 × 6 cylinders, but not for 8 × 8 × 8 cylinders. A problem with 360 volumes (6 × 10 × 10) also failed. The MA18 solver executes 5 × 5 × 5 problems, but appears not to work with a 6 × 6 × 6. The code appears to need more memory. This problem forces the user to use the semi-implicit option for cases with large MULTID components. If running in the “stdy-st” mode, the nearly-implicit scheme must be turned off with the 107 card.		

06035	On-hold Beforehand	BPLU Memory Overwrite	Navy Laboratory Staff
June 2006	In running a system calculation using the 3D hydro (multid) components, the calculation fails in input processing in the BPLU solver reordering. **STATUS** Turning on diagnostics in subroutine BPARAM suggests the arrays are dimensioned too small. The calculation runs with the MA18 solver. This BPLU problem will be addressed during the FORTRAN 95 conversion work. Input deck is proprietary and unavailable.		
06061	**Resolved Beforehand**	**Two Junctions Connect to One TDV**	**Cliff Davis, INL**
November 2006	The code failed while running an EBR-II model. The failure (uses BPLU solver) is caused by connecting two junctions to a tmdpvol. The problem runs if the two junctions are connected to a branch, which has a single connection to the tmdpvol or if the sparse solver (MA18) (Option 35) is used. This problem also fails on PC Versions 236 and 241. This was found in RELAP5-3D Version 2.5.7; this affects all earlier and all later RELAP5-3D versions when the BPLU solver is used.		
07042	**In-work Beforehand**	**Problem w/ Six 3D Components Fails**	**Angelo Frisani, Yassin Hassan, Texas A&M**
October 2007	A calculation with 6 multid (3D hydro) components fails with both the default solver (BPLU) and the optional solver (MA18, Card 1, Option 35). Increasing the FAST common variable 'lfsiz' from 3,500,000 to 22,000,000 allowed the code to run through input processing. The code then fails on the first time step when using either solver. **STATUS** Subroutine TSETSL was modified to increase the scratch space by a factor of 2 for the optional solver (MA18, Card 1, Option 35) as has been done on large problems in the past; this allowed the code to run. Nolan Anderson sent an executable to Angelo and Hassan that contained the FAST change and the TSETSL change for the optional solver. Subroutine TSETSL is currently under investigation with regard to increasing the scratch space for the default solver (BPLU).		
08004	**On Hold Beforehand**	**Insufficient Memory**	**Navy Laboratory Staff**
January 2008	The MA18 (sparse matrix) solver overwrites data in scratch when executing a large problem coupled to a CFD code. The number of coupling junctions (before failure) that can be used with MA18 is more than the number of coupling junctions (before failure) that can be used with BPLU (see UP#06010).		
08025	**In-work Beforehand**	**BPLU Memory Overwrite**	**Navy Laboratory Staff**
January 2008	A PVM problem fails in subroutine EQFINL in Version 2.4.1. The client traced the problem to an issue in the default solver (BPLU) in subroutine BORBND; there appears to be a data overwrite of the variable 'ihld1' in subroutine EQFINL. **STATUS** Client provided the input decks (modification of installation run/PVM directory files pvmcorex.ii, pvmcorep.i, pvmcorec.i to use 6 coupling junctions rather than two coupling junctions). Discussed the problem with Walt Weaver (consultant). Walt noted that this problem needs to be fixed by increasing the scratch space in Version 2.4.1 for the default solver (BPLU); this has been seen previously for both PVM coupled problems and non-PVM coupled problems (see UP#07042). Walt's previous PVM coupled problem studies for the default solver (BPLU) indicate the code runs with two coupling junctions and also fails with nine coupling junctions; this recent problem shows that it fails with six coupling junctions. Walt Weaver suggested installing the code with PVM on and testing the BPLU and MA18 solvers on the six coupling junction problem from a Navy Laboratory. Installed Version 2.4.2 is with PVM on. Ran the client problem with the default solver (BPLU); code failed as it did for the client Reran the client's problem with the optional solver (MA18); code runs. These results are consistent with what Walt saw for the nine coupling junction problem that Nolan Anderson ran. The failure with BPLU is under investigation.		

09024	In-work Beforehand	Problem with 3D Component Fails	Paul Bayless, INL
May 2009	The LOFT L2-5 input deck has two multid components: a 2D component for the downcomer and a 3D component for the core. Using the nearly-implicit solution scheme, attempts were made to run with the 1-D option (CCC0001 word 7 = 1). A code execution failure was encountered at the beginning of the transient calculation with the flag set for just the 3D component. If the flag was set for just the 2D component, or for both components, the calculation ran. This was found in RELAP5-3D Version 2.9.2; this affects perhaps all earlier RELAP5-3D versions, and it affects no later RELAP5-3D versions, since Version 2.9.2 is the latest version. **STATUS** Discussed the failure with Paul; he ran on the PC and there was no traceback. Ran the failed problem on the Unix Dec Alpha and got a traceback; with the BPLU default solver, the failure occurred in subroutine BPARAM (was called from subroutine TSETSL) on the first call to subroutine MAPMAT; with the optional MA18 solver, the failure occurred in subroutine TSETSL on the call to subroutine PMINVD.		
09047	**On-hold Beforehand**	**Problem with 3D Component Fails**	**Peter Cebull, INL**
August 2009	When running the above ATR SBLOCA deck with the default BPLU solver, the code fails. A message generated in subroutine BPPART is written to the screen: "Chain connects to a previous chain." The problem runs to completion using the old solver (Card 1, Option 35). The problem is also seen in Version 2.4.1, so it seems to predate the F90 conversion. This was found in Version 2.9.2; this affects some earlier and all later RELAP5-3D versions.		
10030	**In-work Beforehand**	**Nine Junctions Connect to One TDV**	**George Mesina, INL**
March 2010	In running a modified 3dflow.i installation problem (3dflow.i, that has nine junctions connected to one time-dependent volume), the code fails with a thermodynamic property failure at the minimum time step with both solvers; BPLU (with the fix to UP#06061) fails on vapor case 14 and MA18 fails on Vapor Case 8. All vapor cases with both solvers have large cross flow and take more time steps that the other cases. In running the modified 3dflow.i installation problem on RELAP5/MOD2.5, the code runs all 18 problems with the MA18 solver; the BPLU solver was not available in RELAP5/MOD2.5. For RELAP5/MOD2.5, all vapor cases with MA18 have small cross flow and take the same small number of time steps as the other cases. This was found in a modified RELAP5-3D Version 2.9.4; this affects some earlier RELAP5-3D versions and possibly some RELAP5/MOD3 versions. **STATUS** Currently examining the problem. The deck was created from 3dflow.i.		
11025	**In-work Beforehand**	**Problem with 3D Component Fails**	**Nolan Anderson, INL**
June 2011	The can3d.i problem fails with BPLU but runs with MA18. **STATUS** In all versions of RELAP5-3D that Nolan has constructed, it fails.		

The problems listed as being on-hold were worked on for a while, and some progress was made. The progress was recorded but a solution was not obtained. Problems listed as in-work were actively being debugged. Those listed as resolved were successfully debugged and the updates were incorporated in a version of the code prior to Version 3.0.1.

3. ANALYSIS OF USER PROBLEMS

The UPs were studied and grouped into categories. The input decks for the UP were obtained. The categories and input files for the UP are listed in Table 2.

Table 2. The TestBP input test set broken into categories of BPLU user problems.

UP Number	Name of Input Deck(s)
1. Insufficient memory in pre-transient	
06010	pvmcore12x.i, pvmcore12p.i, pvmcore12c.i
06035	takashi8x8.i
08004	pvmcore12x.i, pvmcore12p.i, pvmcore12c.i
08025	pvmcore12x.i, pvmcore12p.i, pvmcore12c.i
2. Inconsistent reordering in pre-transient	
04021	ap3dsbs.i
04049	pvmcore12x.i, pvmcore12p.i, pvmcore12c.i
06007	lcore6x6x10.i
06013	takashi5x5.i , takashi6x6.i, takashi8x8.i, lcore6x6x10.i
06061	3dflow2.i
09024	l2-5_3d-ni1.i
09047	tran_3in_mod3.i
11025	can3d1.i
3. Error or unphysical result during transient	
99063	UNNEEDED
04052	tht_exe.i, tht_input.i, tht_r5.i
07042	sixmiltid.i, utr07042.i
10030	3dflow2.i

In Table 2, the first category is insufficient memory while executing the pre-transient matrix set-up. This is initiated by subroutine TSETSL calling subroutine BPARAM. The second group is comprised of those runs that quit in the pre-transient because an inconsistent reordering was found while searching for best matrix-shape in BPARAM. The third comprises errors or unphysical results that occur during the transient.

The categories are different from those given in Section 1. There is no unique way to organize them. For example, UP 06035 could also be listed in the insufficient memory category, as could UP 08025 if the category included overwriting of memory at any point in the calculation.

3.1 Analysis of UP not list as solved

There are 4 UPs that were listed as solved at the beginning of the task (underlined and *italicized* in Table 2). The remaining 12 UP are: 04052, 06007, 06010, 06013, 06035, 07042, 08004, 08025, 09024, 09047, 10030, and 11025. The original input decks corresponding to these UP were not all available; the handling of this issue is explained in Section 6.

3.1.1 Analysis of Insufficient Memory in Pre-Transient

UP 06010, 06035, 08004, and 08025 were reported as memory overwrite errors, although 08004 is actually an MA18 overwrite error, but MA18 was used because the number of PVM coupling junctions exceeded the number available with BPLU. Note that input decks for UP 06010 and 08004 were both reported by Navy Laboratory staff and original input decks were unavailable; substitutes were used.

Half of Category 1 problems were solved by conversion to Fortran 95, namely UP 06035 and UP 08025. Conversion to Fortran 95 solved a number of others problems also by providing sufficient storage for BPLU to do its work. The others include UP 04021 and UP 04049. Moreover, errant coding was discovered during conversion and this debugged UP 99063.

The remaining problems are associated with PVM coupling, namely UP 06010 and UP 08004. The PVM Coupling Task had corrected the issues associated with coupling RELAP5-3D to other codes via the PVM Executive as verified by testing during the PVM Coupling Task.

3.1.2 Inconsistent Reordering

An IRUG (International RELAP5 User Group) task to increase code 3D capability was undertaken in 2010, in part to test the new turbulence term. As a result, the size of 3D components is now limited to $99 \times 99 \times 9$. As a result of this task, UP 06007, UP 06013 and UP 07042 are considered solved. Another 2010 task for solving user problems solved UP 06061 by applying graph theory to augment the connectivity matrix of the nodalization diagram's associated digraph. This led to UP 10030.

The remaining problems in this category, UP 09024, 09047, and 11025 are resolved in Section 4.

3.1.3 Error of Unphysical Result in Transient

It was already stated that Fortran 95 conversion solved UP 99063 and that UP 07042 was solved by a 2010 IRUG task for improving 3D capability in RELAP5-3D. This leaves only UP 04052 and UP 10030. These are addressed in Subsections 3.1.4 and 3.1.5.

3.1.4 UP 10030

This problem is considered a physics issue. With either solver, RELAP5-3D/3.0.1 runs all cases when nine junctions are connected to nine separate TDV. When nine junctions are connected to one TDV, the code fails in Case 8 of 3Dflow.i with MA18 and in Case 14 with BPLU.

It is known that these two solvers can produce different results because of their pivoting strategies. (BPLU uses partial pivoting for every step of the decomposition and on every time-step. MA18 uses the same strategy every step until a poor pivot is found.) By pivoting as much as it does, BPLU may be expected to produce a more accurate calculation than MA18 more often. This probably accounts for the code getting through Case 8 with BPLU and not MA18.

This may be a case of ill-conditioning caused by attaching so many junctions to a single TDV. When a linear system has a large condition number, even small differences in the right-hand side can result in large differences in the solution. This affects all linear equation solvers. A poor solution to the pressure-drop equations results in a poor solution to the original system of discretized PDEs.

This is not considered a problem with the linear equation solvers. There are ways of fixing this by modifying the structure of the matrix regarding TDV with multiple connections. This has been submitted as a new UP so that it will be solved.

3.1.5 UP 04052

The problem is an unphysical result because with BPLU and MA18 the code produces different results in a coupled calculation between RELAP5-3D/2.3.6 and CONTAIN/2.0. There is no indication of how different the answers were. The input deck was not reported. However, the input files have been located.

The problem, supplied by Navy Laboratory staff, is a blowdown into the CONTAIN domain which represents the containment. It runs 500 sec to a steady state with 0.1 sec timesteps, then a break occurs until 750 sec. The requested step size is adjusted five times during the blowdown and often exceeds the material Courant limit. With Version 2.3.6, the output differed in the second decimal place for many quantities at 750 sec. For Version 2.3.6 with modifications and subsequent versions that have those modifications, the coupled calculation fails in RELAP5-3D with either solver.

Version 3.0.1 with the MA18 linear equation solver fails at 519.5 sec with a thermodynamic property error at minimum timestep where the variable RHOM is negative in the TDV supplied by CONTAIN. With the BPLU solver, Version 3.0.1 stops at 518 sec because CONTAIN 2.0.2 fails. The results are very close at 500 sec between the coupled calculations with the two solvers. Until that time, there is only heat transfer coupling between the two codes. At 500 sec, the hydrodynamic coupling becomes active. The printed output agrees to 3 or 4 decimal places until the pressure equilibrates and choked flow ends at about 505 sec. Thereafter, the transients proceed in different directions so that by 510 sec, the values are not all the same in the first decimal place.

It is noted with Version 3.0.1, the RELAP5-3D portion of the input model runs, in standalone mode, to completion with both MA18 and BPLU. The results are the same to four significant places, even for quantities on the order of 10^{-12} such as Reynolds vapor.

This indicates that the solvers are producing very similar answers. The fact that small differences in the solution produce large differences in results indicates that either the overall coupled system is ill-conditioned, or there is an error in the coupling between RELAP5-3D and CONTAIN, or both.

Since the three update sets that were added to Version 2.3.6 caused the code to fail with this input deck, they must be examined to determine the cause of the failure in the coupled calculation before any further work, which might prove unnecessary, is undertaken.

4. SOLUTION OF USER PROBLEMS

In this section, the algorithm that resolved the remaining problems, namely UP 11025, UP 09024, and 09047, is presented.

4.1 Source of Remaining Failures

UP 11025, the 3Dcan problem, was the first of the three examined. Diagnostics revealed the source of the failure. During BPLU pre-transient processing, a double loop over BPLU parameters, junction-degree and bandwidth of the banded portion, produces a reordering for each parameter pair. A single pair caused an inconsistent reordering (the matrix would have had a row of zeroes and been singular and therefore unsolvable).

It is necessary to explain the pre-transient processing for BPLU in order to understand the solution. The basis of BPLU is to renumber the unknown to create a matrix with a border-banded structure, or arrow shape. See Figure 1, which represents a simplified vessel with core (1-3), core bypass (4-6), downcomer (7-8), lower plenum (9), and upper plenum (10-11).

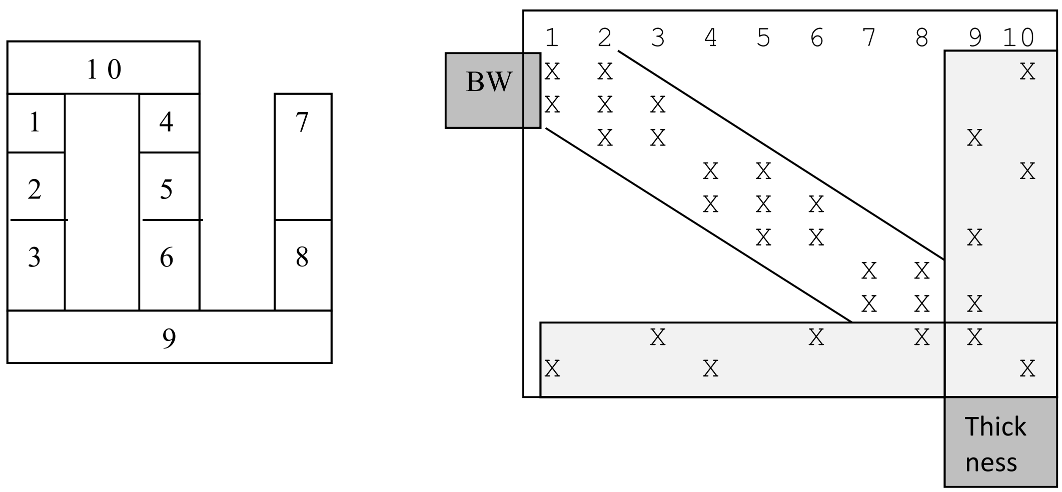

Figure 1. A nodalization diagram and associated border-banded (arrow-shaped) matrix.

The reordering algorithm produces one renumbering of variables for each pair of shape parameters, namely bandwidth (BW) and degree (relates to arrowhead thickness). The degree of a variable is the number of non-zero elements, represented by "X" in Figure 1, in the correspondingly numbered row.

Analysis revealed that in the cases of failure, only one to two of the parameter pairs produced inconsistent reordering while hundreds of pairs produced perfectly good reordering. Two methods were considered for repairing the problem.

4.2 Alternative Solutions

One method to solve the reordering issue is to discern if there are errors in the reordering algorithm and fix them. The second solution is called "Continued Processing." Continued processing ignores the failure condition of a reordering for any single bandwidth value and simply continues seeking values with fully consistent reorderings. It allows the double loop to skip out of the iteration of the inner loop on which a failed reordering occurs; it simply continues processing the next parameter pair.

It was thought that there was no error in the various reordering routines, but that pairs of parameters that do not make sense were being processed. In considering strategy for solving the issue, if errors were sought and none found, the second solution would be implemented anyway. To save time, the second solution was undertaken first.

4.3 Algorithm and Implementation

The subroutines involved are: BPARAM, BPORD, BPPART, BWLIM, and GENCHN. It was necessary to modify these so that the code no longer stopped if an error condition was encountered. Rather, the subroutines set a logical variable that tells BPARAM of the failure. BPARAM was modified to skip out of the iteration of the inner loop if the failure flag was set. It also has a special new failure condition and message if every reordering fails.

Two new variables are introduced:

FAILBW – FAILure in the Band Width iteration loop. Logical variable, true means the reordering with the current bandwidth is inconsistent, false means success.

FAILBP – FAIL in Band Profile Reordering. Logical variable, true means ALL REORDERINGS fail; false means that at least one consistent reordering was found.

4.3.1 Algorithm Modification for Continued Processing

1. FAILBP is initialized to TRUE before entering the outer (degree) loop.
    a. Should FAILBP ever become FALSE, there is at least one good recording.
    b. Subroutines affected: BPARAM, BPORD, BPPART, BWLIM, GENCHN
2. Wherever the fail flag is set, change it to a new BPLU module variable, FAILBW.
    a. FAILBW is initialized to Fail (true) at the top of the outer (degree) loop
    b. Subroutine affected: BPARAM
3. If after a recording routine has done its work, FAILBW is true, then
    a. Subroutine BPARAM combines it with FAILBP via an AND operation.
    b. Thereafter, the current iteration of the inner (bandwidth) loop is exited via a CYCLE statement.
    c. Subroutine affected: BPARAM
4. Collect statistics (degree, bandwidth, operation count and memory requirement) <u>only</u> when FAILBW is FALSE on an inner loop iteration after all reordering routines have finished.
    a. Subroutine affected: BPARAM
5. If, after all parameter pairs are processed, FAILBP is FALSE, inform the user that no reordering can be found to allow BPLU to run.
    a. Subroutine affected: BPARAM.

The modified algorithm should never fail to produce a consistent reordering. *However, in case it does, the user should rerun with the MA18 solver by specifying card 1 option 35.*

Algorithm 4.3.1 was implemented and internal documentation was added to explain operation of the coding. Once all the changes were made and debugged, the 3Dcan problem ran until a thermodynamic property error at minimum time step occurred. This property error also happens with the alternative solver, MA18.Thus the solver issue is considered to be solved.

This bug fix solved UP 11025. Testing revealed that it also solved UP 09024 and 09047.

5. IMPROVEMENTS

Several modifications were made to the BPLU pre-transient source code, new scripts were introduced, and the manual was updated.

5.1 Source Code Improvements

Diagnostics were greatly improved. Diagnostic subroutines were added to output scalar parameters from the two BPLU modules, BPLU1MOD and BPLU2MOD, namely BPLU1MOD_WRITE and BPLU2MOD_WRITE. Further, the graph-theoretic error messages from BPLU were silenced (only available for debugging) as they only served to confuse the user. Subroutines affected: BPARAM, BPORD, BPPART, BWLIM, and GENCHN.

The major reordering algorithms, which were called in the same sequence in two different places from BPARAM, were collected and called from a single internal subroutine. This simplifies and clarifies the main reordering algorithm. Subroutine affected: BPARAM.

To reduce pre-transient processing time, some changes were introduced:

1. When a bad reordering occurs, processing of that bandwidth size stops immediately, this saves a small amount of time. It required that the deallocation of BPLU arrays be performed with a status flag because, prior to this change, *all* BPLU arrays were created for each pass through the bandwidth loop, and now some are not.
2. A test to exit the bandwidth search upon detection that no improvement can be obtained were added. The test is triggered if the width of the arrowhead in Figure 1 ever becomes zero.
3. A new upper limit on the set of bandwidth sizes was introduced. It is calculated from the maximum bandwidth in the original matrix structure.

Of these, the second produces the greatest time savings**.** Subroutine affected: BPARAM.

In the Fortran 77 version, the size of available memory had a fixed upper limit. The solvers were given all space remaining after the RELAP5-3D database was established via input-processing. This established an upper limit space available for the matrix and its auxiliary storage. This space impacted potential BPLU reordering, for if one required too much space, it had to be rejected in favor of one that could fit, but that might have a higher operation count and therefore run slower. The coding for this has become unnecessary with Fortran 90 because memory is allocated as needed. Therefore, it was eliminated. Subroutine affected: BPARAM.

In place of this, now when each BPLU array in the pre-transient is allocated, the status flag is checked and a failure flag is set if there is insufficient memory. The failure flag causes the particular reordering to fail, and the bandwidth size under consideration to be rejected, but then processing continues. Finally, new comments have been added to document the workings of BPARAM in greater detail for future developers. Subroutine affected: BPARAM.

Changes were made to accommodate the pvmcore12x.i deck, which uses a "+" mark in column 1 to indicate that such a line is a continuation of the previous line of input. The variables PROC_MACH, PROC_NAME, PROC_PARMS, and PROC_OUTFILE had to be increased in size. This affected the program units: GETWORD, INPUTD, PROCESSES, PVMCATCHOUT, PVMEXEC, and RR5PVMC.

All changes presented in Sections 4 and 5 are summarized in Table 3.

Table 3. BPLU subroutine changes for bug fixes and improvements.

BPARAM	Documentation explains modified algorithm. Failure flags initialized. Cycle out of inner loop when FAILBW is true after reordering, before statistic collection. Eliminated calculation of statistics for “remaining-memory-fit” reodering. New error message for no matrix reordering has a valid border-banded form. New method and error message for insufficient memory for BPLU matrix solution. New tests for exiting inner (bandwidth) loop earlier to save time.
BPLU1MOD	New subroutine BPLU1MOD_WRITE was created to output module’s scalar quantities.
BPLU2MOD	New subroutine BPLU2MOD_WRITE was created to output module’s scalar quantities.
BPORD	Combination of setting FAILBW flag and return replaces stop statement. No graph-theoretic error messages (except if developer diagnostics active).
BPPART	Combination of setting FAILBW flag and return combination replaces stop statement. No graph-theoretic error messages (except if developer diagnostics active).
BWLIM	Combination of setting FAILBW flag and return combination replaces stop statement. No graph-theoretic error messages (except if developer diagnostics active).
GENCHN	Combination of setting FAILBW flag and return combination replaces stop statement. No graph-theoretic error messages (except if developer diagnostics active).
GETWORD	Sizes of above-listed process module variables were increased. Skip blank lines in input.
INPUTD	Sizes of above-listed process module variables were increased. Process the continuation mark lines.
PROCESSES	Sizes of above-listed process module variables were increased.
PVMCATCHOUT	Sizes of above-listed process module variables were increased.
PVMEXEC	Sizes of above-listed process module variables were increased. Handle fail flag better.
RR5PVMC	Sizes of variables corresponding to above-listed process module variables were increased. Size of line increased from 128 to 2560 and always completely initialized to blank.

5.2 New Scripts and Manual Updates

In addition, several scripts were created or modified.

- New Linux script runBPLU identifies and gives a brief description of each UP tested by a particular test run was built. This is displayed in Appendix A.

- New Linux script runContain runs UP04052 deck coupling RELAP5-3D to CONTAIN.
- New Linux script runTest12 runs the pvmcore12 calculation.
- New Windows makefile Makebp.mak runs all the BPLU UP test cases on the PC.

Scripts were modified to allow the installer to invoke runBPLU if the user creates a file called test in the run directory.

Finally, all RELAP5-3D manuals with references with references to BPLU were revised to modify and slightly correct the wording about BPLU.

6. ADDITIONAL TESTING TO ADDRESS BPLU UP SPECIFIC ISSUES

Testing is required as presented in Section 1. Additional testing was performed as noted in this section.

Tests applied are as follows:

1. Installation Test Set

   These are found in the "run" directory and organized into groups. As the membership of these groups tends to grow, the filenames are listed.

   a. Normal – Tests of basic code features

      2ppumpmod.i, 3dflow.i, 3dflown.i, ans05.i, ans79.i, ans94.i, cstest1.i, cstest2.i, edhtng1.i, edhtng2.i, edhtng3.i, edhtng4.i, edhtng5.i, edhtng6.i, edhtng7.i, edhtng8.i, edhtrk.i, edhtrkd.i, edhtrkn.i, edhtrt.i, edr134a.i, edrst.i, edstrip.i, enclss.i, enclssr.i, fldrn2.i, fwhtr.i, gota27.i, hex2d.i, hex2d1.i, hex2dk.i, hex2dkt.i, hex2dt.i, hex2dt1.i, hstest.i, jetpmp.i, k3200nk.i, marpzd4.i, nc.i, neptunus20., pois_cyl.i, pois_xyz.i, ptkin.i, pump2.i, refbun.i, reflood.i, rk.i, rpump.i, rtsampn.i, rtsampp.i, scw.i, sschf2.i, ssctrl3.i, sstrip1.i, todcnd.i, turbine9.i, typ12002.i, typ1200n2.i, typpwr.i, typpwr2.i, typpwr3d2.i, typpwr952.i, typpwrn2.i, typpwrr2.i

   b. Athena – input decks that feature alternative fluids

      cbdbipb.i, edbh21.i, hxco2.i, iter1.i, pbbi.i, pb_ss_air.i, pipehenxen.i, rcpr.i, scw2.i, vhtrms.i

   c. Other – tests of some special coding

      cmt11.i, cmt11n.i, flecht.i, frigg.i, neptunus17.i, neptunus21.i, one.i, pitch.i, sschf1.i, ssctrl1.i, sstrip3.i, t0301.i, t0301s.i, t0311.i, t0311s.i, tank.i, typ.i, typ3d.i, typ3d2.i, varvol2.i, varvol2r.i, varvol3n.i, varvol5.i, varvol5r.i, vhtrprism.i

   d. Pvm – basic PVM coupling tests with and without the new coupling junction component, See Reference [2].

      cj3wayb.i, cj3wayc.i, cj3wayp.i, cjcore6p.i, cjfailtstb.i, cjfailtstc.i, cjnds.i, cjptc.i, cjpts.i, Def3wayp.i, Defcore6p.i, Defnds.i, Defpts.i, pvmar.i, pvmarc.i, pvmarff.i, pvmarffc.i, pvmarffp.i, pvmarp.i, pvmarwp.i, pvmarwpc.i, pvmarwpff.i, pvmarwpffc.i, pvmarwpffp.i, pvmarwpp.i, pvmcore.i, pvmcore6c.i, pvmcorec.i, pvmcorep.i, pvmcorerc.i, pvmcorerp.i, pvmcs.i, pvmcsc.i, pvmcsp.i, pvmcsrc.i, pvmcsrp.i, pvmDefwayp.i, pvmedac.i, pvmedafc.i, pvmedap.i, pvmedarc.i, pvmedarp.i, pvmedsf.i, pvmedsl.i, pvmff.i, pvmffc.i, pvmffp.i, pvmme.i, pvmmec.i, pvmmep.i, pvmnd.i, pvmndc.i, pvmndr.i, pvmndrc.i, pvmndrs.i, pvmnds.i, pvmnonc.i, pvmnonc1.i, pvmnonc2.i, pvmpt.i, pvmptc.i, pvmptr.i, pvmptrc.i, pvmptrs.i, pvmpts.i, pvmse.i, pvmsec.i, pvmsep.i, pvmvv.i, pvmvvc.i, pvmvvp.i, pvmwp.i, pvmwpc.i, pvmwpff.i, pvmwpffc.i, pvmwpffp.i, pvmwpp.i

2. Developmental Assessment Test Set

   The cases are run to assess code performance against experimental data from separate effects and integral tests. These are listed in Reference [3].

3. DTSTEP Test Matrix

   These input cases test PVM coupling modes and failure conditions. There are about 2800 test cases that are generated from seven basic input files. The generated cases are not listed but are explained in the report, Reference [1].

4. TestDT Test Set

   These input cases test that previously resolved user problems are not affected. These cases are reported in Reference [1].

5. TestBP problem set (BPLU UP Tests) – The input decks listed in Table 2.

All tests are applied on an x86_64 chip with Suse Linux. The non-PVM tests are also run and a Windows XP platform.

6.1 BPLU UP Test Problems

Not all of the input decks corresponding to the UP reports were used for two primary reasons:

- Some contained proprietary data that could not be easily removed by the user. In those cases, input decks were either constructed or others were substituted.
- A substitute that was more complex for the issue under consideration was substituted.
  - For example, it is known that 12 coupling junctions exceed the number in those that failed with BPLU but for which no deck was supplied with the UP.
  - Another example, in place of the EBR-II deck, which had two junctions connected to one TDV, the 3dflow2.i deck was substituted. It has nine junctions attached to a single TDV.

Finally, some UP could not be reproduced. In the case of the FLUENT problem, the FLUENT license is unavailable and exceeds the funds of the project. Further, the user defined functions were never submitted to the archiving system and are unavailable. Also, since both RELAP5-3D and FLUENT have progressed through many versions, they would be out of date and in need of modernization. That task is also beyond the scope of the BPLU task.

The pvmcore12 input deck set is used for three of the UP. It implements a semi-implicit coupling between two RELAP5-3D processes using 12 coupling junctions. This is known to exceed the number of coupling junctions in every UP recorded in Table 1. Therefore, this test set qualifies to substitute for testing the capability of BPLU to handle multiple coupling junctions.

The 3dflow2.i input deck is used to test the solver's ability to handle multiple junctions connecting to a single TDV. It is used in two of the UP listed in Table 1. It was constructed from 3dflow.i by reattaching all nine single junctions that were attached to separate TDV on the right-hand boundary to a single TDV on the right. This test qualifies to substitute for testing the capability of BPLU to handle two junctions as well as nine junctions connecting to a single TDV.

6.2 Results

Test problems were run on both Linux and Windows platforms. For Linux, all the problems in all the categories were run successfully on a Sun Java Station with x86_64 architecture AMD Opetron 64 chips and the Open Suse 2.6 operating system. All problems in TestBp, TestDt and TestMatrix Dt sets of problems run successfully. Among the installation tests, all problems in the normal, Athena, and Other sets run; however, in the PVM test set, four problems that are designed to fail do fail while the rest run. In the DA test set, all problems run successfully except the following five problems terminate with a thermodynamic property failure at minimum time step:

- multi9cyl-ni.i fails near 4.32 sec on advancement 671
- l2-5_3d-ni.i fails near 24.02 sec on advancement 4364
- snc10p1-ni.i fails near 11105.25 sec on advancement 27483
- snc10p4a-ni.i fails near 995.6 sec on advancement 37296
- snc10p4b-ni.i fails near 110.35 sec on advancement 12390.

These expected failures occur with and without the modifications listed above and also in Version 3.0.0. They are not related to a BPLU issue.

For testing on a PC, a DELL Optiplex 755 dual processor (3.00GHz) with Windows XP Professional, Version 2002, Service Pack 3 was used. With the exception of problems requiring PVM, the same input test sets were run on the PC as on Linux. The DA test cases have more problems that on the XP platform than on the Linux platform, as shown in Table 4.

Table 4. DA cases that fail with Version 3.0.1 on Windows XP but not Linux.

DA Case	Fails w/ BPLU	Fails w/ MA18	Succeeds w/ MA18
Rtheta	17 adv	17 adv	
Rigidbody	17 adv	17 adv	
Radial	16 adv	16 adv	
Multi9cyl	14 adv	14 adv	
L2-5_3d	20 adv	20 adv	
L2-5_1d	6332 adv		15709 succeeds
* L2-5_1d_t	14890 succeeds		15432 succeeds

The only difference between using BPLU and MA18 is L2-5_1d which terminates with a thermodynamic property failure for BPLU. This is not a failure of the solver. The transients are the same for the first 15 sec (for example, at 15 seconds, the only difference in the major edit was the value named "err.est" by one in the 7th decimal place) and about the same until 35 sec where the timestep was cut further with MA18 and not BPLU. The trouble occurs near 35 sec. In fact, the following sets of timecards allow RELAP5-3D/ver:3.0.1 to run the transient with BPLU:

```
*          end time  min dt    max dt    optn mnr  mjr  rst
0000201     30.0      1.0-8     0.01     0007  25  500  1000
0000202     35.0      1.0-8     0.001    0007  25  500  1000
0000203    100.0      1.0-8     0.01     0007  25  500  1000

0000201     30.0      1.0-8     0.01     0007  25  500  1000
0000202     35.0      1.0-8     0.00125  0007  25  500  1000
0000203    100.0      1.0-8     0.01     0007  25  500  1000

0000201     30.0      1.0-8     0.01     0007  25  500  1000
0000202     35.0      1.0-8     0.008    0007  25  500  1000
0000203    100.0      1.0-8     0.01     0007  25  500  1000
```

The third set was selected for L2-5_1d_t in Table 4 by recognizing that the Courant limit is slightly in excess of 0.008 for the first 40 sec or so. The difference in code/solver performance arises from different pivoting producing slightly different rounding, resulting in slight differences in the overall solution.

6.3 Conclusions

It has been shown that with the modifications recorded in Sections 4 and 5, BPLU successfully runs all required test cases listed in Section 1, on both Linux and Windows XP platforms. Further, the code runs test cases of all recorded (available) UP from 1999 to 2011 recorded in Table 2, called the TestBp set. Finally, the code runs all the problems in the "Test DT" and "DTSTEP Test Matrix" sets (on a Linux platform). As noted above, the few problems that still fail are due to non-BPLU related issues, usually thermodynamic property errors. User problem reports have been submitted as appropriate to address these issues.

7. REFERENCES

1. G. L., Mesina, "Implementation of a New DTSTEP Algorithm for Use in RELAP5-3D and PVMEXEC Completion Report," INL/EXT-11-20798, January 2011.
2. G. L., Mesina, "PVM Coupling Junction, Software Design, Implementation and Verification Document" INL-CON-10-19917, September 2010.
3. P. D. Bayless, N. A. Anderson, C. B. Davis, P. P. Cebull, E. A. Harvego, "Developmental Assessment of RELAP5-3D Version 2.9.2," INL/EXT-09-15965, Revision 2.

Appendix A

Linux Script that runs BPLU Test Cases

Linux Script that runs BPLU Test Cases

This is the Linux script that runs the BPLU test cases.

```
#!/bin/csh
#  Usage:       runBPLU
#  Description: Test many input decks from BPLU user problems
#  Congnizant:  Dr. George Mesina
#  Created:     August 17, 2011
#
#  1.0  Set up
#
 grep pvmcoupl ../../relap/define > /dev/null
 if ( $status != 0) exit 1
#
#  1.1  Check for executables
 if ( ! -e relap5.x ) then
   if ( -e ../../relap/relap5.x) then
     ln ../../relap/relap5.x .
   else
     echo 'runv:  RELAP5-3D executable unavailable.  Quitting'
     exit 1
   endif
 endif
 if ( ! -x relap5.x ) then
   echo 'runv: No execute permission for RELAP5-3D executable. Quitting'
   exit 2
 endif
 if ( ! -e pvmexec.x ) then
   if ( -e ../../pvmexec/pvmexec.x) then
     ln ../../pvmexec/pvmexec.x .
   else
     echo 'runv:  PVM is specified in the define-file, but\
           the PVM executive is unavailable.  Quitting.'
     exit 3
   endif
 endif
 if ( ! -e pvmcatchout.x ) then
   if ( -e ../../pvmexec/pvmcatchout.x) then
     ln ../../pvmexec/pvmcatchout.x .
   else
     echo 'runv:  PVM is specified in the define-file, but\
           the PVM catchout executable is unavailable.  Quitting.'
     exit 3
   endif
 endif
 if ( ! -x pvmexec.x  || ! -x pvmcatchout.x ) then
   echo 'runv: No execute permission for PVM executive or catchout.
Quitting'
   exit 4
 endif
 if ( ! -e tpfh2o | ! -e tpfn2 ) then
   if (-e ../../fluids/tpfh2o | ! -e tpfn2 ) then
     ln ../../fluids/tpfh2o* .
```

```
     ln ../../fluids/tpfn2 .
   else
     echo runv: Cannot locate property files.
     exit 1
   endif
 endif
#
#  1.4  Pre-cleaning
rm -f *.[pr] >& /dev/null
rm -f *.plt >& /dev/null
rm -f *.out >& /dev/null
rm -f *.log >& /dev/null
if (-d tmp) rm -f tmp/pvmd.* >& /dev/null
#
#  2.0  Run BPLU Test Cases
#
#  UP 04021, 04/2004
#   In version 2.3.2+updates, BPLU failed in pre-transient processing in
#   subroutine BPPART: "chain connects to previous chain."
#
 ./rund ap3dsbs.i
#
#  UP 04052, 10/2004
#   In version 2.3.6, an unphysical result was found wherein heat flux
#   coupling between CONTAIN and RELAP5-3D produced results differing
#   in the first decimal place after unchoking after blowdown with
#   BPLU than with MA18.
#   For code versions after Weaver updates to 2.3.6, the coupled
#   calculation fails before 520 seconds.
#   NOTE: This coupled calculation fails with Version 3.0.1 also. The
#   updates to 2.3.6 are the source and must be reexamined.
#
 ./runContain
#
#  UP 06007, 01/2006
#   In version 3.2.6 with a 3D component, BPLU failed in pre-transient
#   processing in subroutine BPFORM: "inconsistency in reordering, check
#   dimensions."
#  UP 06013, 02/2006
#   In version 3.2.6, BPLU runs 5x5x5 and 6x6x6, but not 6x6x10 or 8x8x8.
#   RELAP5-3D runs with MA18 on 5x5x5, but not 6x6x6.
#  UP 06035, 06/2006
#   In RELAP5-3D 3.2.6, a calculation with 3D hydro failed with BPLU in
#   pre-transient processing. It runs with MA18. Deck proprietary.
#  UP 11025, 06/2011
#   In any versioni with can3d.i, BPLU fails in the pre-transient.
#   NOTE: can3d1.i fails on 3.0.1 at Time=1.27836, adv=60 w/ property
error
#
 ./rund takashi5x5x5.i
 ./rund takashi6x6.i
 ./rund lcore6x6x10.i
 ./rund takashi8x8.i
 ./rund can3d1.i
```

```
#
#  UP 04049, 04/2004
#   In version 2.3.6 with a semi-implicit PVM coupling, BPLU fails.
#  UP 06010, 01/2006
#   In version 2.3.6, BPLU overwrites scratch space when RELAP5-3D is
#   coupled to FLUENT because of too many coupling junctions.
#  UP 08004, 01/2008
#   RELAP5-3D running a large problem coupled to a CFD code overwrites
#   memory with MA18. Cannot BPLU use with so many coupling junctions.
#  UP 08025, 08/2008
#   In a coupled calculation, version 2.4.1 fails in subroutine BORBND
#   by overwriting temporary array IHLD1 in subroutine EQFINL. Source is
#   6 coupling junctions. Decks similar to installation set pvmcore's.
#
 ./runpvm  pvmcore12x.ii pvmcore12p.i pvmcore12c.i
#
#  UP 06061, 11/2006
#   In version 2.5.7, 2.4.1, and 2.3.6, the EBR-II model failed with BPLU
#   but ran with MA18. The source of the error is that the model has two
#   junctions connecting to a TDV.
#  UP 10030, 03/2010
#   In version 2.9.4 with 3dflow.i modified to connect 9 junctions to one
#   TDVi on the right bdy, MA18 fails on case 8 and BPLU fails on case 14.
#
 ./rund 3dflow2.i
#
#  UP 07042, 10/2007
#   In version 2.3.6 running a problem with 6 multid components, the code
#   failed with MA18 unless memory was increased to 22 Mwords. It still
#   failed with BPLU.
#
 ./rund sixmultid.i
 ./rund utr07042.i
#
#  UP 09024, 05/2009
#   In version 2.9.2, a LOFT L2-5 deck with a 2D downcomer and 3D core and
#   nearly implicit with CCC0001 word 7 set for 1D for the core, the code
#   failed. It ran with just the 2D or both regions set for 3D. Failure
#   occurs in the pre-transient with MA18 in TSETSL and with BPLU in
#   subroutine BPPART: "chain connects to previous chain."
#  UP 09047, 08/2009
#   In versions 2.9.2 and 2.4.1 running an ATR SBLOCA with BPLU, a failure
#   occurs in pre-transient subroutine BPPART: "chain connects to previous
#   chain."
#
 ./rund l2-5_3d-ni1.i
 ./rund tran_3in_mod3.i
#
#
#  3.0  Test Improvements
#  The code can now run with 9x99x99 multi-D regions
#
#  3.1 6x6x48
#  It takes nearly 30 minutes to get through input processing on 6x6x48 on
```

```
#  an x86_64 2.2GHz chip and takes 170 sec with BPLU to reach 1.0 sec.
With
#  PGMRES, it takes over 1900 sec.
 ./rund lcore6x6x48.i
#
#  3.2 8x8x16
#  It takes nearly over 6 minutes to get through input processing on
x86_64
#  2.2GHz chip for BPLU and over 7.5 minutes to run to 1.0 sec.
#  With PGMRES, it takes nearly 12 minutes.
 ./rund lcore8x8x16.i
#
#
#  4.0  Clean up
#
#
 rm fort.* >& /dev/null
 rm -f *.plt >& /dev/null
 rm -f *.r >& /dev/null
 rm -f core >& /dev/null
```