

# Release Note



March 3, 2015



## **Differences between 5.0.r246 and 5.0.r215:**

The issues/errors from r215 will be described in some detail so that users will have a better understanding of the differences between r215 and r246. Most of the errors described occur very rarely for unusual models.

### **Issue 1:**

If the environment variable AMLS\_MEM was set to a floating point value, the user would see a message in the .log file like this:

```
/your/amlsexec/path/qd_mod5: line 188: [: 20.0: integer expression expected
```

and the script would then ignore the value that AMLS\_MEM was set to. FastFRS had the same issue.

Fix:

Both the AMLS and FastFRS scripts (qd\_mod5 and fr\_mod2) have been modified to correctly deal with AMLS\_MEM being set to a floating point value.

### **Issue 2:**

For large models with large amounts of area matrix data, phase5 was greatly overestimating how much memory it needed.

Fix:

Memory estimation in phase5 has been improved to consider memory requirements associated with the area matrix more carefully. As a result, the memory needed in phase5 for very large models is reduced dramatically.



### Issue 3:

Phase2 was improperly assessing whether it could reuse an existing mapping as indicated by the AMLS\_P2\_MAP environment variable. This resulted in the following message:

The substructure tree definition from the file "amls5\_p2\_map2" does not fit this model, so AMLS is defining a new substructure tree.

The problem with the previous definition is:  
Unable to find a node that must be in the last ss in the list of input marked nodes

This was caused by phase2 incorrectly determining which nodes were required to be in the last substructure when it evaluated the validity of the list provided in amls5\_p2\_map2.

Fix:

Phase2 has been corrected to properly determine which nodes must be in the last substructure when evaluating the list in amls5\_p2\_map2.

### Issue 4:

There was an error in phase3 that produced output like this:

```
*****Error in AMLS Phase3*****  
Error: B and C non-conformal  
Function: admatrix::gemm(A,ad_B,ad_C)  
*****  
Called by: p3_ss::form_msl  
Called by: p3_ss::msl_aux  
.  
.  
.  
Called by: p3_ss::msl_aux  
Called by: phase3
```



Called by: fork function for AMLS Phase3  
Called by: main amls driver

This error occurred when a node related to a potential rigid body mode or mechanism needed to be moved to the last substructure, and that node was the only one coupling a substructure to its parent. This violated expectations in the remaining phase3 computation. This error was extremely rare, and has only ever been encountered on one unusual model.

Fix:

When a case like this is encountered, the substructure tree is now redefined to properly represent coupling between substructures.

#### Issue 5:

There was an error in phase5 that produced output like this:

```
working on volatile case
input blength is 12077208
needed_bytlength is 12077216
largest space is 8722080
total length is 750027520
mode is 1
Volatile is 1
Normal is 0
```

```
*****Error in AMLS Phase5*****
Error: unable to find requested space in work_item
Function: work_item::get_space(bytlength)
*****
Called by: workspace::get_space(bytlength)
Called by: matrix<T>::point_to(workspace &work, i_t nrow, i_t ncol)
Called by: blmatrix::allocate(mask)
Called by: p5_ss::expand_phi
.
```



.  
.  
Called by: p5\_ss::expand\_phi  
Called by: handle\_phi  
Called by: phase5  
Called by: fork function for AMLS Phase5  
Called by: main amls driver

Fix:

A bug in a function that sized memory space for phase5 was fixed.

#### Issue 6:

AMLS results when reusing a mapping in phase2 were not always as similar as they should have been. This was caused by the fact that the second run used its own criteria to select nodes for the last substructure, which may not have resulted in the same set of nodes in the last substructure as in the first run, even though the base mapping used was the same.

Fix:

Phase2 has been improved to write both the mapping and the list of nodes to put in the last substructure in the file amls5\_p2\_map2. The second run then attempts to use both. If either the mapping or the list of nodes to put in the last substructure is not valid, neither is used and phase2 informs the user that the mapping from the previous run was not reused.

#### Issue 7:

AMLS could produce a memory error if one of the specified output degrees of freedom was not in the model.



Fix:

Both the amls translator and phase2 now check to make sure all specified output degrees of freedom actually are in the model before attempting to use them.

#### Issue 8:

In the AMLS translator, the process of making sure that the input system matrices were symmetric was very slow for large models whose mass matrices were nearly as dense as their stiffness matrices.

Fix:

The algorithm for making sure that the matrices are symmetric has been greatly improved.

#### Issue 9:

AMLS file usage statistics have been over-counting disk usage for some files.

Fix:

Statistics on the maximum filespace used by AMLS should now be accurate.

#### Issue 10:

One of the system matrices in a certain model was not populated symmetrically: an  $(i,j)$  value was very close to zero, but the corresponding  $(j,i)$  value was actually zero. This resulted in a glibc memory error in phase2.

Fix:

The AMLS translator has been improved so that it detects this type of



problem and corrects it.

**Change 1:**

AMLS user output sent to the Nastran .log file, related to licensing issues in particular, has been improved.

**Change 2:**

In some linux distributions, ksh (the Korn Shell) is no longer included by default, so all AMLS scripts have been changed to use bash (the Bourne-Again Shell).

**MSC and NX Delivery Data Bases (DDB):**

On the CDH AMLS FTP site the following DDBs are stored:

MSC 2010.1.3, 2011, 2012.1, 2012.2, 2013.1, 2013.1.1, 2014, and  
NX8.5, NX9.0, NX9.1, NX10.

If there is a special need for an older DDB version, please contact:

[support@cdh-aq.com](mailto:support@cdh-aq.com).