SOFTWARE ARCHITECTURE AND HIERARCHY OF THE NASA MULTISCALE ANALYSIS TOOL

Evan J. Pineda[1], Trenton M. Ricks[1], Brett A. Bednarcyk[2], Steven M. Arnold[3]

**1. Research Aerospace Engineer, 2. Research Materials Engineer, 3. Technical Lead: Multiscale Modeling
Multiscale and Multiphysics Modeling Branch
Materials and Structures Division
NASA Glenn Research Center
Cleveland, OH USA**

SUMMARY

The NASA Multiscale Analysis Tool (NASMAT) serves as a state-of-the-art, “plug and play,” software package which utilizes multiscale recursive micromechanics as a platform for massively multiscale modeling for hierarchical materials and structures subjected to thermomechanical loads on high performance computing systems. This paper is intended to give an overview of the design of NASMAT and how the design supports modularity, upgradability and maintainability, interoperability, and utility. First, the software architecture and hierarchy will be explored. Details on each of the 11 NASMAT procedures and the arrangement of NASMAT data will be presented. Finally, application program interfaces (APIs) that were developed to facilitate the communication of NASMAT with other programs will be described.

1: Overview of NASMAT

The design of the NASA Multiscale Analysis Tool (NASMAT), written primarily in Fortran 90, has focused on modularity, upgradability and maintainability, interoperability, and utility. The various operations of the code are compartmentalized into a set of 11 procedures: Pre-processing, Driver, Engine, Solution, Homogenization, Material Model, Localization, State Change, Output, Post-processing, and Utilities. Each procedure has access to a library of modules. Specific modules can be swapped in and out, as needed, to solve the multiscale problem of interest. Numerous procedures, and the subroutines used within the modules, are recursive. Moreover, recursive data types are used extensively to handle the large quantities of data associated with each length scale considered by the multiscale model. The recursive nature of the NASMAT procedures and data enabled the development of a thermomechanical, multiscale recursive micromechanics (MsRM) framework capable of supporting massively multiscale modeling (M^3) on high performance computing systems (HPC). Finally, application program interfaces (APIs) have been developed to support communication between NASMAT and other external programs.

2: NASMAT Procedure Hierarchy

To facilitate the previously described functionality objectives for NASMAT, the necessary computational operations were categorized into 11 procedures. Those procedures include *Pre-processing*, Driver, Engine, Solution, *Homogenization*, Material Model, *Localization*, *State Change*, *Output*, *Post-processing*, and Utilities. The *italicized* procedures indicate procedures that must be recursive to support multiscale modeling. The hierarchy of these procedures is given in Figure 1. The function of these procedures along with the current, and possible future, modules available for use in each procedure will be described in the following subsections. Note, the Utilities and *Post-processing* procedures do not fall within the NASMAT hierarchy because they are ancillary.

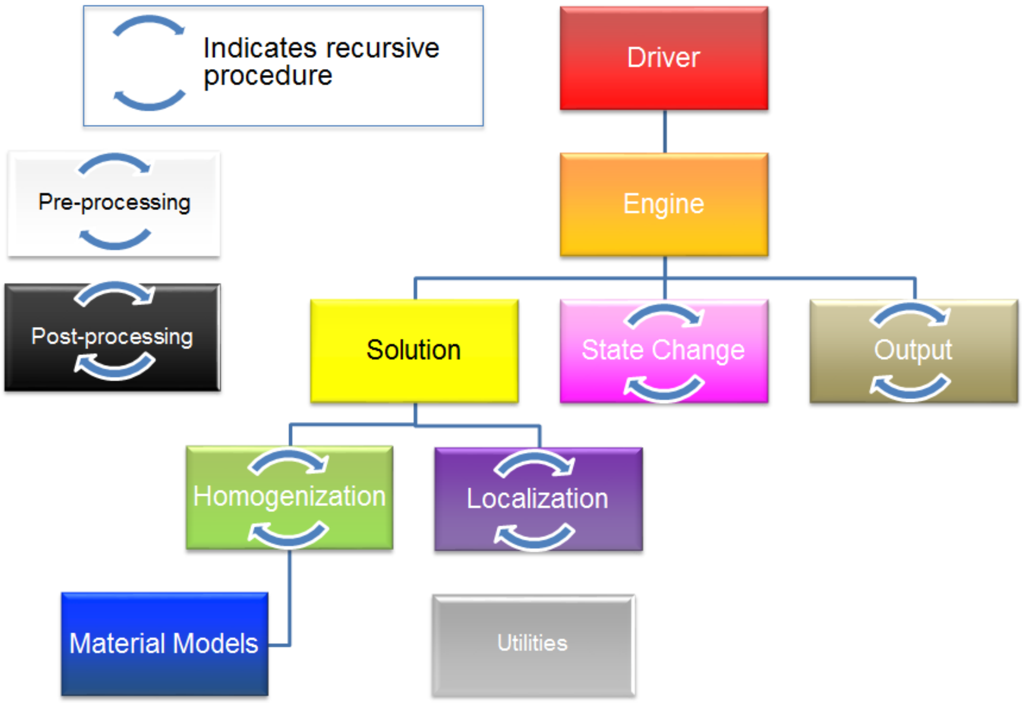

Figure 1: Hierarchy of NASMAT procedures. Each procedure has access to a library of modules that can be exchanged to solve the particular multiscale problem of interest. The curved arrows indicate that the procedure is recursive.

Pre-processing: In this ancillary procedure all the necessary input files are read including the NASMAT `.mac` file and any files need for user defined modules. The NASMAT data structures are also allocated in this procedure. The data allocation must be recursive to support MsRM.

Driver: The Driver is used to provied the global thermal and mechanical load increments. The currently available Driver modules include user-defined, quasi-static load, inegration within Abaqus finite element method (FEM) software, and integration with any other third-party structural analysis software through the NASMAT MacroAPI.

Engine: The Engine is the core of the NASMAT software package and serves as the interface between the Driver and Solution procedures. The Engine is unique and does not contain any modules. All NASMAT analyses must integrate the engine into the workflow.

Solution: The Solution dictates the appropriate solution method needed to solve the problem of interest. The current Solution modules in NASMAT are the constiuent solution, which is used to solve problems involving monotlithic materials, and the micromechanics solution, which is used to analyze heterogenous materials defined using a repeating unit cell (RUC) with periodic boundary conditions. Although not currently availabe, a classical lamination theory solution module will be implemented in NASMAT version 2.0 [1].

Homogenization: In order to obtain the necessary infromation to solve a non-linear thermomechanical problem, the Solution procedure must call two other procedures. The first is the *Homogenization* procedure. The modules within this procedure include various micromechanics theories including the generalized method of cells (GMC), the high fidelity GMC (HFGMC), the parametric HFGMC (PHFGMC), the Carerra unifed formulation (CUF) and the Mori-Tanaka mean field theory (MT) [2-5]. The *Homogenization* procedure provides the Solution procedure with the effective thermomechanical properties of the RUC considering material nonlinearity in the RUC constituents. This procedure must be recursive to enable MsRM. CLT homgenization, and homogenization utilizing user-defined micormechanics, through the NASMAT MicroAPI, will be available with NASMAT version 2.0.

Material Models: The *Homogenization* procedure must have access to the Material Models procedure in order to obtain the constitutive response of the material points in the RUC. The currently available Material Models include linear elastictiy, linear thermal expansion and material point elimination (MPE). With MPE, all of the componets of the stiffness matrix for the material point are reduced to a near zero value after satisfaction of a failure criterion which is evaluted in the State Changes procedure. Progressive damage and viscoplastic modules will be added to the Material Models solution for NASMAT version 2.0.

Localization: In order to solver a nolinear micromechanics, or multiscale, problem, the Solution procedure must also invoke the *Localization* procedure. It is required that the theory used in the *Localization* procedure is complimentary what is used in the *Homogenzation* procedure. The *Localization* procedure is responsible for calculating the local fields (stress, strain, etc.) within the RUC. This procedure must be recursive to support MsRM.

State Changes: The NASMAT Engine calls the recursive *State Changes* procdure to determine if the state of the material changes and set the appropriate flags. These flags will be utilized in Material Models procedure to elicit the appropriate constituent response. Currently, only material point failure criteria are supported within the *State Changes* procedure, but other possible *State Changes* modules include crack growth, oxidation, microstructural changes, and interfacial debonding.

Output: The recursive *Output* module writes the desired output data to the hard disk. Currently data can be output in an ASCII text-based format, or into a hierarchical HDF5 database.

Utilities: There are several ancillary utility functions that are made available throughout the entirety of the NASMAT program.

Post-processing: The *Post-processing* procedure is handled outside of the NASMAT analysis hierarchy and involves visualization of the NASMAT data. It is necessary that this procedure is recursive to enable multiscale visualization.

3: NASMAT Data Management

To utilize shared-memory multi-processing programming, such as Open Multi-processing (Open MP) global data variables cannot be used to store local, independent data from each core. Moreover, M^3 requires the storage of a tremendous number of independent variables. Therefore, to exploit HPC and to organize the data for MsRM, the data used in NASMAT is categorized into 7 types: Deck, Fields, Properties, Flags, Step, Solution, and RUC Data. Derived type data structures are defined for each of the data categories, and these data structures are passed by reference throughout the code. The data structures are themselves recursive. I.e., a member within a particular structure is of the same derived type as the parent structure. This allows for seamless MsRM framework in which the same data variables, and the same recursive subroutines, can be used for computations at any scale, including the global. The data types are categorized as follows:

Deck: Deck contains all of the information read from the `.mac`, or user-defined, input files. The Deck data structure is static upon application and population and remains as a reference.

Fields: Fields are the quantities which culminate into the observed deformation of the RUC. NASMAT Fields include stress, strain, and temperature, for example.

Properties: The Properties data structure contains all of the material property information needed to invoke the appropriate Material Models solution module. Examples include material stiffness matrix, compliance matrix, coefficients of thermal expansion (CTEs), and failure criteria allowables.

Flags: Flags are data, usually binary, that indicate outcome of a Boolean operation within NASMAT. These flags are initialized in the *Pre-processing* solution and are updated, as needed, throughout the code. An example of a Flag datum the failure of a material point. This Flag is false until a failure criteria associated with that material point is satisfied.

Step: This data structure contains information on the global time incrementation and non-linear solution iterations. Since, all of this data is global, and NASMAT does not yet support disparate time scales or nonlinear solution procedures, this structure is not recursive.

Solution: All of the data that is needed in the Solution, *Homogenization*, and *Localization* procedures modules to complete the desired operations is stored in the Solution data structure. An example of Solution data is the strain concentration matrix in GMC [2].

RUC Data: RUC Data contains all of the geometric data pertaining to the RUCs in the analysis. It also contains indicators of the materials that comprise each constituent.

4: NASMAT Application Programming Interface

Several NASMAT API are under development to support user defined procedure modules within NASMAT. These API are written to support Fortran, C, and C++ interfaces. The two API that are near completion and will be available in NASMAT version 2.0 are MacroAPI and Micro API. The MacroAPI integrates NASMAT within a third-party structural analysis software, such as an FEM solver, to provide material point properties. The MicroAPI integrates a user-defined micromechanics theory within the NASMAT MsRM framework. The MicroAPI is still under development and will not be presented in this paper, but two external micromechanics theories (PHFGMC and CUF) have already easily been integrated as modules within NASMAT [3, 4].

MarcroAPI: Figure 2 contains a diagram outline of the user-defined procedures that must be deployed in order to interface NASMAT with a third party structural analysis code to provide material point information. The uncoloured shapes with dashed outline represent operations that must be defined by the user. The actual NASMAT interface that is used is dependent on the language that the higher-level code is written in (Fortran, C/C++). The MacroAPI has already been used to interface the Abaqus, CalculiX, and the internal General Electric (GE) Scalable Paralell FEA (SPAF) FEM codes. Figure 2 is partitioned into three separate user-defined procedures.

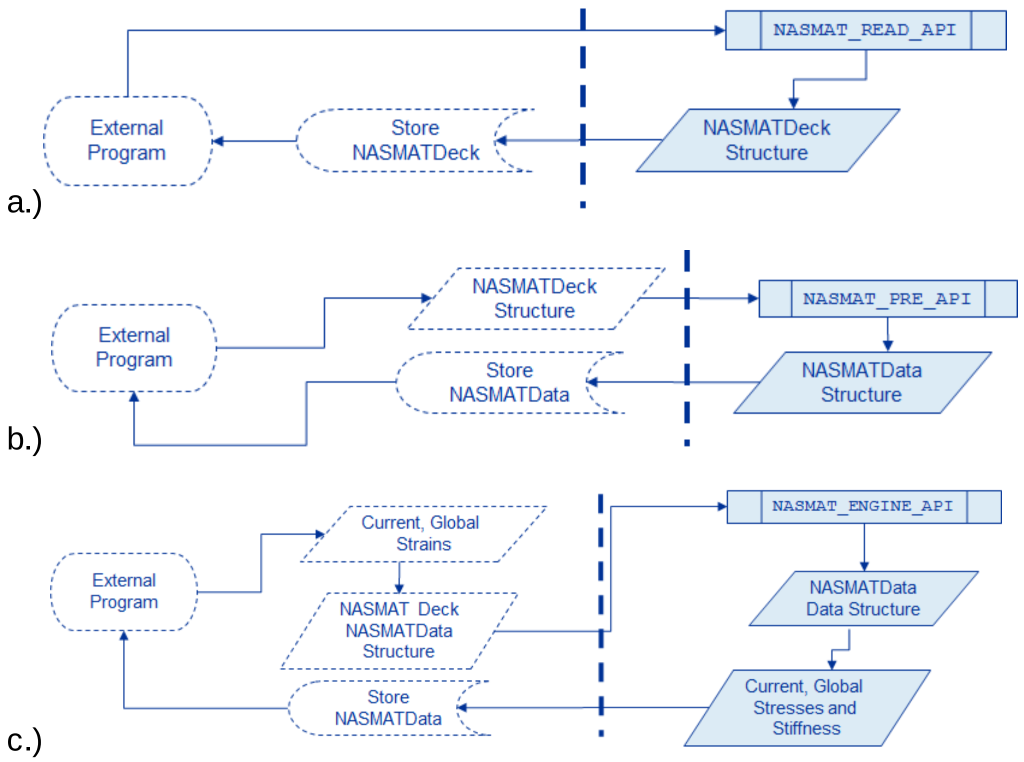

Figure 2: Diagram show three necessary user defined procedures in the NASMAT MacroAPI which interfaces NASMAT with external, structural analysis codes. The uncolored dashed shapes indicate that the operation must be defined by the user. a.) NASMAT input file reading. b.) NASMAT pre-processing. c.) NASMAT analysis.

First, shown in Figure 2a, the user must call the front end subroutine for the *Pre-processing* module that reads the NASMAT input files:

NASMAT_READ_API. NASMAT will then allocate and populate the NASMATDeck data structure and pass it back to the user subroutine. The user is responsible for storing this data structure and passing it back unaltered to other MacroAPI procedures. This user-defined procedure must be implemented in the user code for each unique NASMAT input file needed in the analysis.

Next, the user must call subroutines that perform NASMAT data allocation and initialization through NASMAT_PRE_API, as shown in Figure 2b. The user must pass the NASMATDeck data structure, associated with the current user-defined material point, to NASMAT_READ_API, and NASMAT_READ_API will return the NASMATData data structure which contains all of the other data structures described in Section 3 as children, except Deck. This user-defined procedure must be invoked for every user-defined material point that is to be linked to a NASMAT analysis.

Finally, the user must run the NASMAT analysis for each desired material point via NASMAT_ENGINE_API. The user must pass the current material point strain, and the appropriate NASMATDeck and NASMATData structures associated with this material point. NASMAT_ENGINE_API will return homogenized thermomechanical properties of the relevant RUC, including stiffness and CTE, as well as the global volume averaged stiffness that is work conjugate to the strains that were provide by the user.

5: Conclusions

NASMAT is intended to be a virtual platform for simulating the thermomechanical response of heterogeneous materials and structures. This was achieved by developing an MsRM framework which utilizes recursive subroutines and data structures. To facilitate “plug and play” interoperability with other software as well as modularity, upgradability, maintainability, the NASMAT hierarchy is categorized into 11 key procedures which are described in detail in this manuscript. Each of these procedures has access to a library of modules which can be used to solve the specific problem of interest. The NASMAT MacroAPI was developed so that NASMAT can be easily integrated, through a Fortran or C/C++ interface, into external structural analysis codes for multiscale simulation of structures.

NASMAT version 2.0 is scheduled for release in October, 2020 and will include modules to support the capabilities described in this manuscript. However, beyond 2020 the NASMAT developers will continue to build the libraries of procedure modules to include plasticity, user-defined materials, interfacial debonding, fatigue driver and damage, message passing interface (MPI) parallelization, evolving microstructures and multi-physics capabilities.

ACKNOWLEDGMENTS

Primary development was funded through the NASA Aeronautics Research Mission Directorate's (ARMD's) Transformational Tools and Technologies (TTT) Project. Additional support came from ARMD's Advanced Composites (AC) Project and the Space Technology Mission Directorate's Composite Technology for Exploration (CTE) Project. The authors are also grateful to Dr. Ibrahim Kaleel, Dr. Marco Petrolo, and Dr. Erasmo Carrera for help, guidance, and assisting with the integration of CUF into NASMAT and similarly, Dr. Rami Haj-Ali for integrating PHFGMC2D within NASMAT.

REFERENCES

1. Jones, R.M., 1999. *Mechanics of Composite Mateirals, Second Edtion*, Taylor and Francis, Inc., Philadelphia.
2. Aboudi, J., Arnold, S.M. and Bednarcyk, B.A., 2012. *Micromechanics of Composite Materials: A Generalized Multiscale Analysis Approach*, Butterworth-Heinemann.
3. Haj-Ali, and R., Aboudi, J., 2013. A new and general formulation for the parametric HFGMC micromechanical method for two and three-dimensional multi-phase composites, *International Journal of Solids and Sturctures*, 50, pp. 907-919.
4. Carrera, E., Maiarú, M., Petrolo, M., and Guinta, G., 2013. A refined 1D element for the structural analysis of single and multiple fiber/matrix cells, *Composite Structures*, 96, pp. 455-468.
5. Mori, T., and Tanaka, K., 1973. Average stress in matrix and average elastic energy of materials with misfitting inclusions, *Acta Metallurgica*, 21, pp. 571-574.