Project Information

Company Name: RNET Technologies
Title: VnV: A Self Documenting Testing Framework for In-situ Verification and Validation in High Performance Computing Applications.
Principal Investigator: Dr. Ben O'Neill
Sub-Contractor: Dr. Gregory Watson (ORNL)
Topic Number/subtopic letter: 30d (Modeling and Simulation)

Problem Statement

Verification and Validation (V&V) of numerical simulations is a discrete process that cannot realistically account for every possibility. This causes issues in general purpose simulation packages because, while it is the responsibility of the developers to ensure the product is mathematically correct, it is the responsibility of the end-user to ensure the solution is a suitable representation of the physical model. After all, the direct costs of a design failure, be it time, money or loss of life, fall squarely on the shoulders on the end-user, and any attempt to shift the blame to simulation library X will certainly fall on deaf ears.

General Statement

The VnV toolkit facilitates the development of *explainable* numerical simulations that, in addition to the final solution, provide the end-user with a detailed report on why the solution can be trusted.

Phase I Feasibility

The Phase I effort demonstrated the feasibility of developing a framework that facilitates end-user V&V in advanced numerical simulations. This included the development of a portable solution for declaring injection points in a code, an intuitive interface for writing tests and injecting them at runtime, and an automated documentation generation system with support for 2D and 3D visualization.

Phase II Plans

The Phase II effort will develop the complete VnV framework, including the development of custom preprocessor directives that allow injection points to be configured in more dynamic ways and efficient statistical metrics for asserting the state of data stored in distributed arrays. The value of the VnV framework will be demonstrated through integration into MOOSE, PETSc, and libMesh.

Commercial Applications and Other Benefits

Numerical modeling and simulation (M&S) is almost always more economical than live prototyping; a fact that has seen wide-scale uptake of M&S in industry (e.g., automotive, nuclear, aerospace, advanced manufacturing, etc.). In all cases, the explainable numerical simulations facilitated by the VnV toolkit provide end-users with the wealth of knowledge required to ensure dangerous errors do not propagate into final designs.

Key Words

Verification, Validation, Accreditation, In-situ testing, HPC, Documentation generation, ADIOS, NEAMS.

Summary for Members of Congress

The VnV framework provides scientists with the tools required to create explainable numerical simulations that, in addition to the final report, provide users with a detailed report on why the solution should be trusted. Such information will help end-users ensure uncaught simulation errors do not propagate into final designs.

30d (Modeling and Simulation)
DOE FY18, Release 2 Phase II SBIR/STTR

Phase II Proposal

VnV: A Self Documenting Testing Framework for In-situ Verification and Validation in High Performance Computing Applications
Phase I Contract Number: DE-SC0018728

Ben O'Neill[1], PI   Gregory Watson (Sub-contractor)[2]

[1] RNET Technologies
240 W Elmwood Dr
Dayton, OH 45459-4296
boneill@RNET-Tech.com

[2] Oak Ridge National Laboratory
1 Bethel Valley Road MS6016
Oak Ridge, TN 37831
watsongr@ornl.gov

Contents

1 Significance, Background, and Technical Approach

RNET Technologies Inc. (RNET) in Dayton, OH and Oak Ridge National Laboratory (ORNL) are responding to the 2019 DOE SBIR/STTR Phase II Release 2 (DE-FOA-0001976). This proposal is for a Phase II contract in succession to an initial Phase I contract (Contract #: DE-SC0018728) awarded for DOE SBIR/STTR Topic 30d (Modeling and Simulation). Based on a prototype developed in Phase I, RNET and ORNL are proposing the development of VnV: a self-documenting testing framework for end-user solution verification and validation in high performance computing applications. This proposal will highlight the need for, and the tremendous value of, the proposed tool in high performance numerical simulations. The goal of the proposed Phase II project will be to develop and demonstrate a production quality implementation of the VnV framework that delivers this value across a broad spectrum of numerical applications.

RNET has extensive SBIR experience in high performance computing, including performance optimization of numerical software and libraries, development of fine-grained power monitoring tools for HPC infrastructure, and the development of software usability tools that enhance the user experience. Dr. Watson, our collaborator at ORNL, is an experienced software professional with extensive knowledge of software design, architecture, and engineering practices. He has experience developing parallel debugging software for HPC environments and is the project leader of the open-source Eclipse Parallel Tools Platform (PTP). We believe that our experience developing advanced numerical software for the HPC community combined with Dr. Watson's experience with scientific software engineering uniquely qualifies our team to develop and commercialize the VnV framework.

1.1 Identification and Significance of the Innovation

1.1.1 Identification and Significance

Numerical modeling and simulation (M&S) is almost always more economical than live testing and prototyping; a fact that has seen wide-scale uptake in the R&D life-cycles of products ranging from $10 polycarbonate toys, up to solar panels, airplane wings and nuclear reactors. With access to high performance computational resources at an all time high, and with exascale computing resources on the horizon; the role M&S has in the design pipelines of next generation technologies is only expected to increase. However, numerical simulations are, by definition, an *approximation* to a real world physical system. As such, it is important that this increased reliance on simulated tests is accompanied by a concerted effort to ensure simulations are fit for the intended purpose. As stated in the DoD best practices guide [2], verification and validation (V&V) of a code should be performed when *"...the risk of making an incorrect decision based on a simulation outweighs the time and cost of performing V&V to ensure that a simulation produces results that are sufficiently accurate and reliable."* One only needs to look to the Sleipner platform accident[7, 6], where an offshore oil platform collapsed due to failures in the finite element simulation, to get an idea of the devastating consequences poorly verified numerical simulations can have on business, the environment, and society.

The staples of a rigorous V&V regimen are:

- The development of a detailed V&V plan.
- The implementation of software development best practices (e.g. version control, unit and regression testing, code reviews, etc.).
- Mathematical and algorithmic testing (convergence analysis, mesh refinement studies, method of manufactured solutions, etc.).
- Development of a broad benchmark testing suite.
- Uncertainty quantification and sensitivity analysis.
- Comparison of simulation results with experimental data and results from third party simulations.
- Review of the implementation and results by experts in the field.
- Documentation of the V&V effort.

V&V is a discrete process that cannot account for each and every possibility. This raises issues in the development of general purpose numerical simulation packages because, while it is the simulation software developers responsibility to ensure the product is mathematically correct, it is ultimately the responsibility of the end user to ensure the solution is a suitable representation of their physical model. After all, the direct costs of a design failure (be it time, money, or loss of life) fall squarely on the shoulders of the end-user; any attempt to shift the blame to the developers of simulation library X will certainly fall on deaf ears.

End-user V&V is particularly important in the DOE Nuclear Energy Advanced Modeling and Simulation (NEAMS) program. NEAMS is developing predictive models for the advanced nuclear reactor and fuel cycle systems using leading edge computational methods and high performance computing technologies [4]. The NEAMS group has placed a significant emphasis on V&V [3]; the NEAMS tools have integrated functionality for input validation (through the workbench and MOOSE), mesh refinement and method of manufactured solution analysis (through MOOSE), and uncertainty quantification and sensitivity analysis with DAKOTA (through the workbench). Despite this, the complex nature of the codes combined with the high stakes nature of nuclear reactor design has created a need for tools that automate solution verification for end-user driven simulations.

To address this need, RNET Technologies Inc. (RNET) and Oak Ridge National Laboratory (ORNL) are proposing the development of the VnV framework; a C/C++ software package that facilitates end-user V&V in general purpose numerical simulation packages (e.g., MOOSE, PETSc, libMesh, Fenics, OpenFoam, etc). The framework will promote the development of *explainable* numerical simulations that, in addition to the traditional simulation solution, produces a detailed report outlining how the solution was obtained and why it should be trusted. The VnV framework will provide simple to maintain V&V to create self verifying, self describing, explainable numerical simulations.

1.1.2 Product Overview and Technical Approach

In this proposal, we make the distinction between the *V&V of a simulation package* and the *end-user driven V&V of a solution obtained using a simulation package*. These two processes are not independent; indeed, the V&V of a numerical simulation package almost always includes a set of verified and validated benchmark tests; likewise, the assertion that a package is verified and validated forms the foundation of trust in solutions obtained by end-users. The focus of the Phase II project will be end-user validation; however, the functionality imparted by the framework can be used in the V&V of simulation packages as well.

The VnV framework will facilitate the development of explainable numerical simulations through:

- **In-situ Testing And Analysis:** Unit tests are an effective mechanism for ensuring a function works as expected. However, unit testing is an unavoidably discrete process that cannot cover every possible outcome. This is particularly true for numerical algorithms because even small changes (e.g., input parameters, mesh geometry, etc.) can cause the algorithms to behave unexpectedly (i.e., diverge,

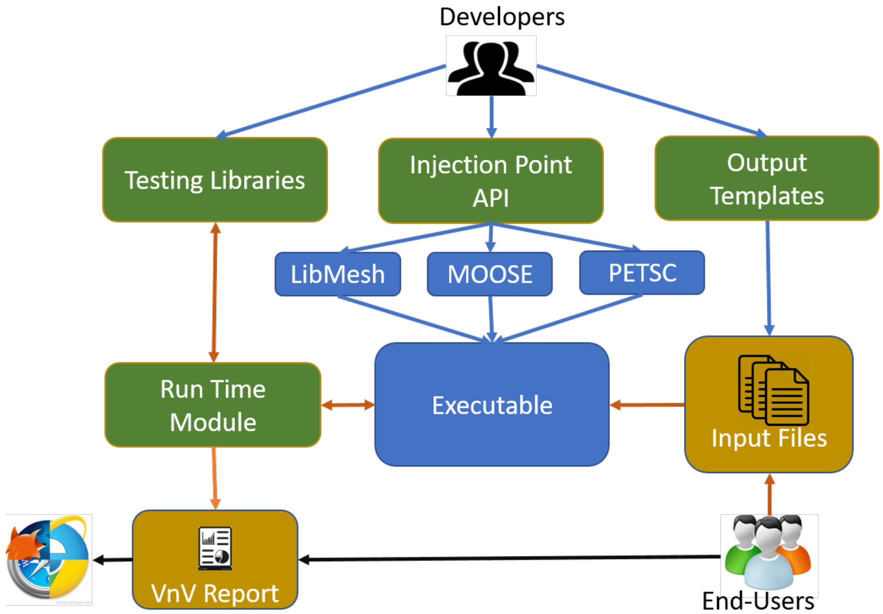

Figure 1: The VnV toolkit. Here, green boxes represent core functionalities. Developer interactions are shown in blue, runtime interactions are shown in orange and post-processing interactions are shown in black.

converge to the wrong solution, etc.). As such, a robust V&V report should include a description of unit tests completed *and* a detailed set of tests and assertions that were completed during the simulation process.

The VnV framework will include a sophisticated test injection system with cross-library support for defining testing points in existing codes. The framework will be able to configure injection points in any library linked to a simulation. For example, a MOOSE user would be able to run V&V tests at injection points defined in the source codes of hypre, PETSc, libMesh, MOOSE, etc through a single interface. This cross library support will allow for in-depth, expert directed, end-user V&V in executables that utilizes a range of numerical simulation libraries.

- **Reusable Software Components:** While the specific details of V&V vary from application to application, the macro scale algorithms are relatively consistent (e.g., mesh refinement studies, the method of manufactured solutions, sensitivity analysis, uncertainty quantification and error propagation). Many of these algorithms can be, or already have been [1], implemented as black-box or near black box solutions. The VnV framework will provide a robust set of near black-box tools that implement these common V&V approaches.
- **Efficiency:** Performing V&V tests in a distributed environment will be expensive, both computationally and due to the data movement required to support generic domain decompositions. The VnV framework will offer functionality for offloading tests to external processes. This will significantly reduce the run time costs of completing V&V using the VnV framework.
- **Documentation Generation:** With software packages under almost constant development, and new and improved packages being released on a regular basis, keeping an up-to-date V&V report is an almost impossible task. The VnV toolkit will include automatic VnV report generation in the form of a server-less HTML web page. The report will be built using an extended markdown format with support for standard markdown formatting, latex formatting, images, videos, self-sorting tables, two-dimensional charts, and three-dimensional visualization.

Figure 1 uses the MOOSE tool-chain to show how developers and end-users will interact with the VnV framework. The first step is to define the injection points. These injection points will be placed at key locations of the code where testing can and should take place. Developers will also complete an output template describing the state of the simulation at each injection point. That specification will be used to populate the final VnV report.

The next step is to create a VnV test. The tests are developed in external libraries and hence, can be developed either by the developer of the simulation or by the end-user of the library. The core framework will also include a robust set of general purpose V&V tests. Each test will be accompanied by a markdown formatted template file. Like injection points, this markdown file will be used to describe the test and present the results. The VnV framework supports a custom markdown format that includes a range of data visualization techniques. We envision that the developers of a numerical simulation package will ship the library with hard-coded injection points and a set of custom V&V tests.

End-users will be able to generate a customized input configuration file for each executable. This configuration file will contain information about every injection point located in the call-graph of the simulation; including those in external 3rd party libraries. After customizing that file, generating a VnV report is as simple as running the simulation.

In summary, once integrated into an application, the VnV framework will provide a simple mechanism for creating self verifying, self describing, explainable numerical simulations. This will significantly reduce the burden associated with V&V for end users, thereby increasing the usability of the tools for non-expert end-users. The core functionality of the VnV toolkit will be released as open source for use in academic and enterprise applications, with RNET providing commercial support, training and integration contracts to interested parties.

1.2 Anticipated Public Benefits

Numerical modeling and simulation (M&S) is extensively used in numerous scientific and engineering fields and disciplines such as computational fluid dynamics, high energy physics, nuclear engineering, and computational finance. The prevalence of smaller scale finite element analysis (FEA) software is even wider, with applications ranging from design optimization in $10 polycarbonate trinkets up to large scale parameter optimization of fighter jet wings. In all cases, the software used to inform the designs of these products must be verified and validated; and in all cases, the verification and validation of this software is a complex, time consuming task requiring input from experts across the broad spectrum of numerical simulation specialty areas (i.e., linear solvers, nonlinear solvers, finite element methods, physical domain specification, data analysis, etc.).

End-User V&V is essential, expensive to setup and maintain, and supporting end-user V&V is difficult. Undetected errors in numerical simulations propagate into physical designs creating issues that can cost millions of dollars to fix, cause catastrophic damage to the environment, and, in the worst case scenarios, result in the loss of human life. While most high quality simulation packages have robust internal V&V regimens, few (if any) ship with a functionality that streamlines end-user V&V . Instead, software simulation developers often take a legal approach, whereby the license includes language to the effect of *use at your own risk* or *this software is released as-is, with no guarantee it is fit for the intended purpose*. The detailed VnV reports do not shift the burden of end-user V&V to the developer, that is, and always will be the responsibility of the end-user; rather, the framework provides a mechanism for providing the end-user with a level of knowledge about the inner workings of the simulation far beyond what is provided in most numerical simulation tools.

The customers of the VnV framework include the commercial and government entities that develop large-scale numerical models and simulations (e.g., ORNL, Idaho National Laboratory, AFRL, ANSYS, CD-adapco, universities). For these customers, RNET will provide training, support, and integration services to integrate the VnV framework into new and existing code bases, as well as contract based services for extending the toolkit to fit specific needs. For these customers, the benefits of the VnV framework include; explainable numerical simulations that increase trust in the solution (VnV enabled simulations produce both a traditional result and a description of how the solution was obtained and why it can be trusted), runtime V&V configuration (reduces setup time and allows the users of the code to iteratively build a robust V&V regimen without ever touching the source code of the simulation), a robust collection of V&V tools (improves efficiency by providing statistical assertions optimized for performance in a large scale distributed settings), automated production grade documentation generation (reduces time to market and reduces the burden of meeting the V&V reporting requirements), and cross-library support (multilingual, cross library V&V testing provides a simple mechanism for facilitating end-user V&V at all levels of the simulation hierarchy). While focusing on end-user V&V, the features provided by the VnV framework benefit both simulation developers (our customers) and end-users (users of the simulation software). Additionally, including end-user V&V features into their products will provide a differentiating feature to their end-users.

The true beneficiaries of the V&V framework are the end users of the advanced numerical simulation products. By reducing the burden associated with end-user V&V , the VnV framework will afford these researchers and engineers with the knowledge required to drive the next generation of technological advancements.

1.3 Phase I and Feasibility Demonstration

The goal of the Phase I effort was to demonstrate the feasibility of a framework that facilitates end-user solution verification in advanced numerical simulation. To that end, the project team spent the majority of the Phase I project developing a prototype of the VnV toolkit.

1.3.1 The VnV Injection Point System

The injection point system was written in C++ with a focus on minimizing the amount of code needed to insert injection points into existing code bases.

Integrating of an injection point into an existing code is a three step process; (1) include the header file, (2) place injection point in the code and (3) write the output template. Figure 2 shows a function that has been augmented with a single stage injection point. Developers insert injection points using the INJECTION_POINT macro. The format for this macro is:

```
INJECTION_POINT( <name>, <stage>, <type> <variable>, ...)
```

Here, “name” is the id that will be used to define the injection point in the configuration files and the final reports. This name must be unique across all injection points in an executable. The stage parameter is an integer value that defines the step that this injection point belongs to. Using this parameter, the developer can set up multi-staged injection point testing. This allows for data collection across multiple code locations.

The developer defines the variables available for inspection using the variable parameter. During preprocessing, the macro expands this variable definition as

```
<type> <variable> --> "type", (void*) &variable
```

At runtime, the VnV injection point factory compares the “type” of each variable against the type of variables supported by each VnV test. When a match is found, the framework casts the void* pointer back to its correct type and forwards it to the test for processing. There are several risks associated with this approach, the most important of which is the memory corruption the could occur from a bad cast. Addressing these issues is a major goal of the the Phase II project (see Section 2.2).

The final step in the creation of an injection point is to write the output template. This template is used to populate the final VnV report. The output template is a YAML file containing a markdown formatted description of the injection point. There is no requirement that a output template be supplied, however, the more information that can be provided, the more informative the final VnV report will be. The VnV framework supports a custom markdown format that includes a number of data visualization extensions (2D charts, 3D visualization, Tables, latex, etc). Most importantly, the custom markdown format provides a mechanism for automated post-processing of the data collected during VnV testing.

```
class LinearFunction {
public:

    double slope, intersection;

    LinearFunction(double slope_, double intersection_) :
      slope(slope_), intersection(intersection_) {}

    double eval(double x) {
       double value = slope*x + intersection;
       INJECTION_POINT("IP_1",-1,
                       double slope,
                       double intersection,
                       double x,
                       double value);
       return value;
     }
};
REGISTER_IP("IP_1",-1,double slope,double intersection,
             double x, double value);
```

Figure 2: A code Snippet showing a member function enhanced with a single stage injection point called “IP_1”. This injection point is declared inside a member function designed to evaluate a linear function at a particular value of x.

1.3.2 The VnV Testing Interface

The second facet of the injection point system are the V&V tests. The test interface was developed under the assumption that tests should be loaded at runtime and defined independently of the source code. As such, the test interface was built using a C++ plugin pattern.

The framework includes a library generation script that will automatically build the directory structure and makefiles required to build a testing library. Once the library has been initialized, the user can begin to develop individual tests.

To assist in the development of tests, and to avoid issues with incorrect type-casting, a test generation script is also included. This script automatically generates the boiler plate code required to implement the testing interface while also taking care of the required typecasting. With this, implementing a custom test requires the developer to:

- **Implement the "declareParameters" function**. This function defines the parameters required to perform the test. For example, the test in Figure 3 requires two doubles; one called slope and the other called intersect. The user will map the injection point variables to the test variables using the test configuration file.
- **Implement the "runTests" function**. The variables passed to the test are direct pointers to the variables in the code. Hence, tests should not modify the pointers in any way; however, the Phase I prototype does not strictly enforce this requirement. Beyond that requirement, there is not limit on the procedures that can be run inside a test. All data required for post-processing should be output inside this function using the supplied ADIOS engine.
- **Write the output template.** Like injection points, all tests should be accompanied by a output template. The output can be used to automatically post-process and visualize the data collected during each test.

1.3.3 The Runtime Module

Once the injection points have been declared and the tests created, the next step is to develop the XML test configuration file. Using this XML file, users can configure which testing libraries to load and which tests to run. The test configuration file is used to map injection point parameters to test parameters. For example, Figure 4 shows a portion of a test configuration file that inserts the LinearTest function shown in Figure 2 into the "IP_1" injection point defined in Figure 3. Please see the final report for a full description of the XML format supported by the Phase I prototype.

Turning on VnV testing across all libraries linked to the executable is as simple as calling an initialization function with a valid input configuration file.

1.3.4 Automated Report Generation

The final step in the VnV pipeline is the automatic report generation. At the core of the report generation system is a custom markdown extension. This extension supports a range of custom data visualization features that interact directly with the output data in the ADIOS2 output file. For example, users can specify markdown that renders as an interactive 2D visualization automatically populated with data obtained during VnV testing (the line chart shown in Figure 5 is generated using this functionality). Other features include 3D visualization using VTK.js and search-able tables with tabular.js. The extension also includes support for running custom python scripts during compilation. This allows for infinite possibilities with regard to processing the test outputs. For example, a user can write markdown that, during compilation, runs a paraview script and then display the results using the VTK.js.

Figure 5 shows a screen-shot of a VnV report generated using this approach. The main layout consists of two components; the index and the content. The index in generated directly from the VnV output file. Each entry in the index represents an injection point that was reached during the execution of the simulation. The content section is generated automatically from the YAML specification files. In this case, the

```
#include "injection.h"
#include <math.h>

class LinearTest : public IVVTest {

    bool valid = false;

    void declareParameters(std::map<std::string,std::string> &parameters) {
        parameters.insert(std::make_pair("slope", "double"));
        parameters.insert(std::make_pair("slope", "intersect"));
    }

    // The actual testing code. In this case we check if the slope and intersection
    // point are valid (>0) and write the values to file.
    TestStatus runTest(adios2::Engine &engine, int stage, double* slope, double* intersect) {

        valid = ( slope > 0 && intersect > 0 ) ? 0 : ( slope <= 0 ) ? -1 : 1;
        engine.Put("slope", *slope);
        engine.Put("intersect",*intersect);
        engine.Put("valid", valid);
    }

    void declareIO(adios2::IO &io) {
        io.DeclareVariable<double>("slope");
        io.DeclareVariable<double>("intersect");
        io.DeclareVariable<int>("valid");
    }

    // BoilerPlate Code Automatically generated by the test generation script.

    TestStatus runTest(adios2::Engine &engine, int stage, NTV& parameters ) {
        double* d0 = carefull_cast<double>(stage,"slope", parameters);
        double* d1 = carefull_cast<double>(stage,"intersect", parameters);
        int testStage = m_config.getStage(stage).testStageId;
        return runTest(engine, testStage,d0,d1);
    }
}

//More boiler plate code generated by the test generation script. This code registers
// the test with the runtime module when the shared library is loaded.

extern "C" {
      IVVTest* LinearTest_maker(VVTestConfig &config) {
        return new LinearTest(config);
    }
  void LinearTest_DeclareIO(adios2::IO& io) {
     LinearTest::DeclareIO(io);
  }

};

class LinearTest_proxy {
public:
    LinearTest_proxy(){
        // Register the test with the factory
        VV::test_factory["LinearTest"] = std::make_pair(LinearTest_maker,LinearTest_DeclareIO);
    }
};

LinearTest_proxy lt_p;
```

Figure 3: An example of a custom VnV test. In this case we implement a test that checks that the slope and intersection point of the linear function are positive and writes the result to file.

```xml
<ns1:testLibrary>
  <ns1:path>../tests/build/testLib.so</ns1:path>
</ns1:testLibrary>

<ns1:injectionPoint name="IP_1" markdown="injection_points.yaml">
 <ns1:test name="LinearTest" markdown="linear_test.yaml">
  <ns1:testStage ipId="-1" testId="1">
   <ns1:parameter from="slope" to="slope"/>
   <ns1:parameter from="intersection" to="intersect"/>
  </ns1:testStage>
 </ns1:test>
</ns1:injectionPoint>
```

Figure 4: An example of the XML configuration file for attaching test to injection points at runtime. In this case, we attach the LinearTest test shown in Figure 3 to the ”IP_1” injection point shown in Figure 2.

user has completed a test titled “Linear Function Plotter” inside an injection point called “Linear Function Constructor”.

The key point to note here is that this interface was generated automatically from the VnV output file.

In summary, during Phase I, the project team created a functioning prototype of the VnV framework that provides:

- A clean mechanism for inserting injection points in existing codes
- A simple interface for defining custom tests
- A report generation system that automatically creates an interactive server-less web report based on the VnV output.

As will be described in the work-plan, the goal of the Phase II project will be to take this initial prototype and extend, harden and optimize it for use in high performance computing applications.

2 The Phase II Project

2.1 Technical Objectives

The requirements being addressed include the development of a robust framework for in-situ verification and validation in general purpose numerical simulation packages. In particular, the objective of this project is to address the need for tools that automate verification of end-user numerical solutions in the NEAMS toolkit. The Phase II project will meet this need by developing a framework that facilitates the creation of explainable numerical simulations that produce a final solution *and* a detailed report on why that solution should be trusted.

Toward this goal, RNET Technologies and ORNL will pursue the following objectives:

1. {Harden and extend the prototype developed during Phase I. In particular, the Phase II effort will focus on developing more robust mechanisms for specifying injection points inside existing code bases. }
2. {Develop mechanisms for efficient in-situ comparisons between data in a distributed environment. In-situ comparison of variables with experimental and/or analytical results will significantly reduce the amount of IO required in V&V testing, while also providing a fine grained mechanism for detecting at what point a solution diverges from the expected result.}

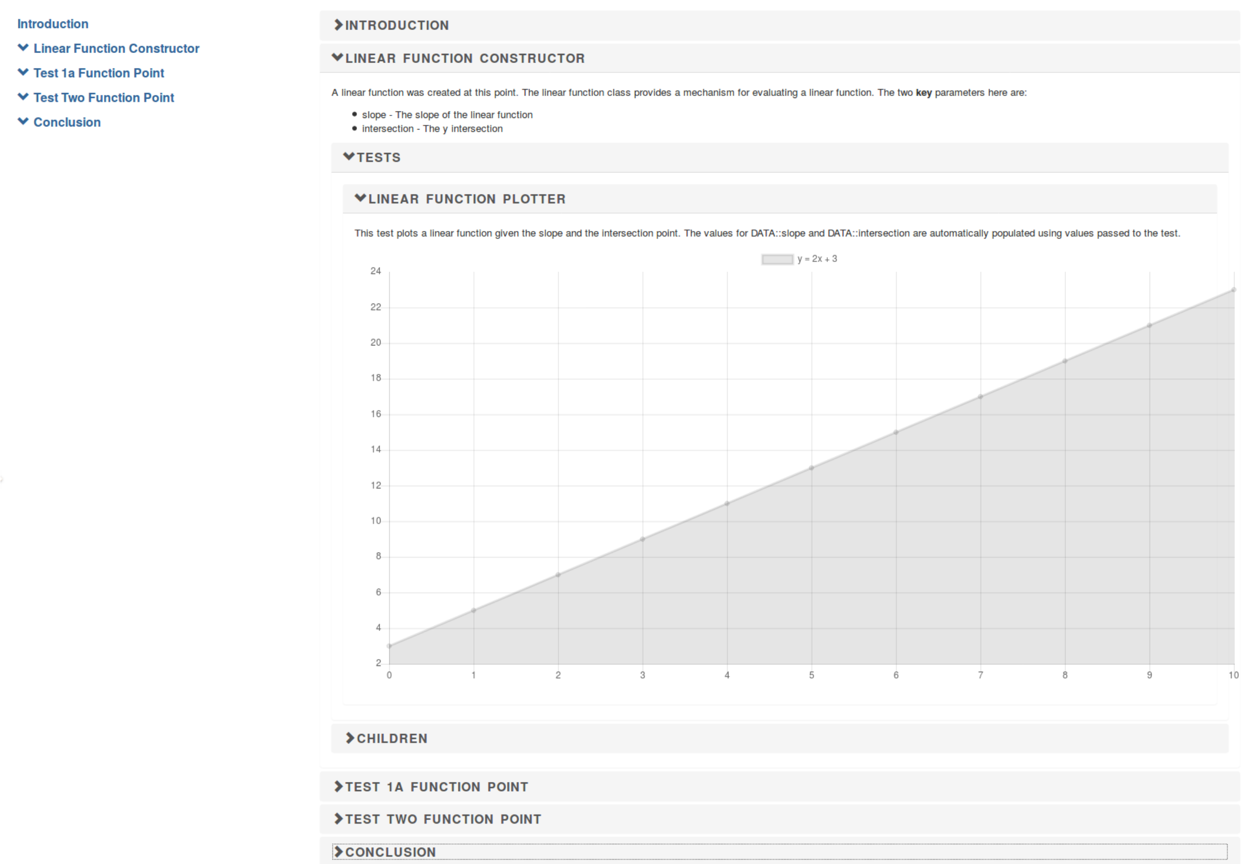

Figure 5: An example of the VnV report generated automatically from the VnV output files and the injection point and test specification files.

3. {Minimize execution times through job based parallelism. This will shift the burden of V&V testing out of the main simulation, allowing for faster runtime, especially for testing methods that require the same function to be executed multiple times.}
4. { Develop a robust set of generic V&V tests. The development of these tests (e.g., mesh refinement studies, the method of manufactured solutions, sensitivity analysis, uncertainty quantification) will equip the users with the tools required to perform end-user V&V .}
5. {Demonstrate the value of the VnV framework as a component in the NEAMS tool-chain (MOOSE, libMesh and PETSc). Integration into these tools will answer the NEAMS call for tools that support end-user verification of numerical simulations. Integration into the NEAMS workbench will provide access to the tools in the seamless manor users of the workbench have come to expect. }

2.2 Work Plan

The goal of the Phase II project will be to develop a fully functional, efficient, battle tested framework for integrating advanced end-user verification and validation into general purpose numerical simulation packages. In what follows, we outline the work required to achieve this goal.

2.2.1 Hardening and Optimization of the Injection Point System

The injection point system developed during Phase I relies on C style Macros and string based pointer casts. In a *trusted* environment, this is an efficient (void* casts are almost free) and portable (it is written in C) approach. However, there are three main weaknesses need to be addressed prior to production:

- **String based pointer casts:** The system developed in Phase I uses developer provided strings to infer the type of each variable. Under the hood, the injection point system uses these strings to ensure compatibility between test parameters and injection point variables. The key issue is that the strings specified at each injection point cannot be verified during compilation. This will causes issues in cases where, say, the developer changes the type of a variable, but forgets to update the type string in the injection point declaration.
- **Restrictive type specifications:** The current system uses C compliant pre-processor macros to simplify the process of describing injection points and variables. The benefit of this approach is that the injection points can be compiled into any application without significant changes to the build system. The downside is that the single-pass, text based pre-processing supported by the C preprocessor places a significant restriction on the functionality that can be imparted.
- **Constant Correctness** The current system provides the testing algorithms with direct access to the data structures and variables specified at each injection point. This access allows for efficient, unrestricted testing of the data structures. There are some exciting use cases for this functionality (parallel debugging, computational steering, etc.); however, from the perspective of V&V, it is imperative that testing routines do not alter the trajectory of the simulation in any way.

In Phase II, the project team will investigate and develop an annotation based system for defining injection points. This annotation system will provide users with full control over what variables can be accessed at each injection point, including the ability to provide access to internal components of data structures, describe the domain decomposition of distributed arrays and to complete pre-test processing. The annotation system will also provide a mechanism for injection point detection in non-object orientated programming languages where runtime injection point detection cannot be completed. However, the primary benefit of this new system will be that it will automatically detect the type of the variables tagged for inspection at each injection point.

To implement this annotation system, the project team will create a set of custom pre-processor directives (i.e., pragmas) using Clang. Clang is a compiler front end for the C family of languages. It is a well supported, well documented compiler package designed as a drop in replacement for GCC. We we utilize Clangs library API for implementing custom Pragma routines to implement a pre-processor that transforms the annotations into valid injection point specifications.

To address the issues with constant correctness, the team will develop support for fine-grained regression testing. This functionality will allow users to track the progress of a simulation against a VnV output file that was generated without in-situ testing. This will allows users to compare the current state of the simulation against an expected result at every level of the simulation.

2.2.2 Efficient Statistical Comparisons in Distributed Settings

One of the major benefits of the VnV framework is that the tests are executed inside the simulations distributed environment. This allows for the development of efficient, parallel testing algorithms. To capitalize on this functionality, the Phase II effort will include an investigation into efficient statistical methods for analyzing, asserting and comparing data stored in distributed arrays.

The first step toward working with distributed arrays is to obtain some information about the global partition of the data. This information is readily available in simulations that use regular data decompositions (i.e., block, cyclic, etc.), but it is not generally known in simulations where the data is distributed irregularly across the processors.

The naive approach to determining the global partition is to explicitly form the partition using global MPI communication. This is a robust approach that will be immediately applicable in a large number of situations. However, the large storage cost required to build the global partition (O(P), where P is the number of processors) is likely to become an issue in peta- and exa-scale environments [5].

To that end, the Phase II effort will include a detailed analysis of the optimal algorithms for forming the global partition in a distributed setting. This will include the implementation and profiling of the aforementioned naive approach, as well as an investigation into more advanced approaches parallel rendezvous algorithms such as forming a hash based distributed directory [9] and the assumed partition algorithm [5].

The second step will be to develop efficient algorithms for comparing and analyzing distributed data based on that partition. This will include the development of the VnV Testing suite for distributed Arrays. This testing suite will include a variety of analysis routines for analyzing data including, means, std deviation, variance, co-variance, etc. A collection of functions implementing matrix based metrics will also be included (e.g., one norms, symmetry, positive definiteness, etc).

A key goal of the Phase II effort will be to develop support for comparing solutions obtained in distributed arrays with data stored on disk. This will be achieved using the ADIOS2 read-write API. In this case, the global partition determined above will be used in conjunction with ADIOS to distribute the correct data to each processor. The project team will also perform a detailed analysis of existing third party data transfer and interpolation tools to determine the best approach to efficiently comparing experimental data with the data in the distributed arrays. The project team will also investigate the feasibility of implementing a multi-cast reduction network to perform reductions and comparisons on the data.

2.2.3 Reducing Run-times with Job Parallelism

Performing V&V tests in a distributed environment will be expensive, both computationally and due to the data movement required to deal with the domain decomposition employed by the application. To address this problem, the VnV toolkit will support the offloading of tests to external processes. This functionality will allow for speedup through job parallelism for expensive testing routines like sensitivity analysis and mesh refinement, while also allowing for specific tests to be run on an optimal architecture (e.g., a test involving image processing could be offloaded to a GPU enabled architecture).

The first step toward supporting test offloading will be to implement an effective mechanism for transferring the required data from the simulation to the external testing processes. To do this, the project team will use the ADIOS 2 Sustainable Staging Transport (SST). The SST is a classic streaming data architecture that allows for direct connection between data produces and data consumers.

The VnV toolkit will use the SST engine to stream the required testing data to a VnV Testing Manager. The test manager will determine the best available resource for running the given test and launch the job. To do this, the test manager will utilize the extensive support for integrating with remote job schedulers available in the Eclipse Parallel Tools Platform (PTP).

The PTP project provides an integrated development environment to support the development of parallel applications written in C, C++, and Fortran. Eclipse PTP provides; support for the MPI, OpenMP and UPC programming models, as well support for a wide range of batch systems and runtime systems, including PBS/Torque, LoadLeveler, GridEngine, Parallel Environment, Open MPI, and MPICH2.

In this case, PTP provides a simple interface for launching tests as separate processes. The open research question to be answered during the Phase II effort is to determine in which situations test offloading is beneficial. In particular, the project team will look to implement heuristic algorithms that detect when the cost of streaming the required testing data outweighs the performance benefits associated with offloading the tests.

2.2.4 Integrate Third Party Tools for Mesh Refinement, UQ and Sensitivity Analysis.

Mesh refinement studies, uncertainty quantification (UQ) and sensitivity analysis (SA) are all essential components of a robust V&V regimen. To that end, the Phase II effort will include the development of a VnV testing library that integrates with third party tools to facilitate testing using these approaches.

UQ and SA will be developed using DAKOTA. DAKOTA provides a set of black-box tools for performing parameter optimization, UQ and SA. The Phase II effort will include the development of a flexible interface for setting up and running DAKOTA tools at injection points, and the development of the output templates for displaying the results. Most high performance finite element packages support some level of adaptive mesh refinement (e.g., libMesh, Fenics, MFEM, etc.), hence, rather than trying to patch in third party tools, the mesh-refinement tool will be developed as a generic interface for interacting with existing mesh refinement functionality.

2.2.5 Integration with NEAMS tools and the NEAMS workbench

To demonstrate the value of the VnV framework, the project team will integrate the VnV framework into the open-source components of the NEAMS tool-chain (PETSc, libMesh and MOOSE). Doing so will provide ample opportunities for testing, while also allowing us to demonstrate the performance of the toolkit in real codes with real applications. Moreover, this integration will provide a road map for integration into MOOSE based NEAMS applications like BISON.

Integration into NEAMS tool-chain will be a three step process:

- The first step will be to insert injection points into key locations in the MOOSE, libMesh and PETSc source codes. The optimal locations for inserting these injection points will be determined after detailed profiling of example codes; however, some options include during each linear solver iteration, during each nonlinear iteration, during finite element matrix construction (if it exists) and inside the function for calculating the action of the Jacobian on a vector (for matrix free problems).
- The second step will be to allow users to configure the VnV testing directly in the MOOSE input file using the MOOSE input file syntax. This will provide users of MOOSE with a seamless mechanism for setting up and running tests. This will involve writing a custom MOOSE action for setting up and configuring the VnV runtime module.
- The third step will be to enable context-aware auto-complete for MOOSE based VnV configuration files in the NEAMS workbench. The workbench has integrated support for extracting the information required to setup input file verification and auto-complete from MOOSE applications; however, there will likely be some additional work required to correctly setup this up for tests stored in external libraries. In particular, the current system requires that the tests adhere to a specific XSD specification, but there is not yet a system for extracting the exact parameters required to inject a specific test.
- The last step will be to provide support for viewing the final VnV reports in the NEAMS workbench. The project team demonstrated how the QWebEngineView can be used to display the HTML VnV report in a QT window. The Phase II effort will extend that demonstration by providing the callback function required to interact with the NEAMS workbench interface (menus, docks, etc).

2.3 Performance Schedule and Task Plan

RNET would like to present the project ideas and research plan to the DOE Program Manager and other interested scientists. The meeting will be used to discuss features and to identify the specific NEAMS applications and computer resources that will benefit from this project. This meeting will be scheduled soon after the Phase II contract is awarded. The meeting can be hosted at RNET, a DOE site suggested by the Program Manager or via a teleconference.

RNET will submit all reports as required by the contract (e.g., annual reports, a continuation report, summary reports, and a final report) to the DOE program manager and other interested DOE scientists.

The research and development topics described in Section 2.2 will be addressed by the tasks described in the remainder of this section. Most tasks require active collaboration between RNET and ORNL. Figure 6 summarizes at a high level the dependencies among tasks and approximate anticipated task durations. The duration of the Phase II project is 104 weeks. Specific details are included in the description of each task.

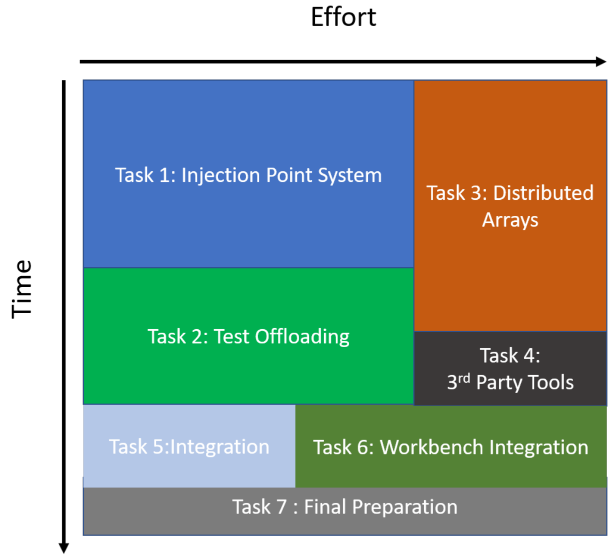

Figure 6: Overview of task dependencies and timeline.

2.3.1 Task 1: Development of the Annotation based Injection Point System

{In this task, the project team will develop the Annotation system for specifying injection points and describing injection variables. This will include the development of the custom LLVM compiler extensions required to process these annotations. At the end of this task, users will be able to specify injection points and injection point variables using either the new Annotation point system, or using the more portable, but less robust Phase I approach.

To test the Annotation based injection point system, the project team will write injection points at several key locations in the MOOSE software stack, including inside the source code of several MOOSE modules, the core MOOSE framework, libMesh and PETSc. This will allow us to demonstrate the cross-library potential of the Annotation based system, while also acting as the first step toward integration of the tools a number of the NEAMS tools.}

{ RNET will work on the implementation for this task and ORNL will provide inputs and guidance. }

2.3.2 Task 2: Develop methods for offloading tests to external processes.

{In this task, the project team will add support for test offloading using the ADIOS2 SST architecture. This will include the development of the VnV test manager and integration with the PTP platform, a task which will likely involve custom modifications to the PTP platform. Once implemented, test offloading will be tested using the VnV enabled MOOSE software stack developed in Task 1. The goal of this task will be to see a significant speedup when using test offloading over the standard in-situ testing routines. }

{ RNET will work on the implementation for this task with ORNL providing assistance with ADIOS and any modifications to PTP. }

2.3.3 Task 3: Development of efficient statistical V&V tools for distributed arrays.

{In this task, the project team will investigate the optimal approaches for conveying information about the data decomposition. This will involve robust analysis of three approaches to determining the global partition; explicitly forming the global partition; building a distributed directory; and using an assumed partition algorithm. This task will also include the development of a number of efficient statistical tools for asserting the state of the data in distributed arrays.

At the completion of this task, users will be able to efficiently compare and analyze distributed arrays using the VnV framework. To test these implementations, the project team will set up tests for performing assertions on various PETSc vectors and matrices. This testing will compare the performance of each of the global partition methods and each of the analysis tools.}

{ RNET and ORNL will collaborate on this task. }

2.3.4 Task 4: Development of Generic tools for Mesh refinement, Uncertainty quantification and Sensitivity Analysis.

{In this task, RNET will implement generic V&V tools for mesh refinement, uncertainty quantification and sensitivity analysis. In the case of mesh refinement, the approach taken will be to create a generic interface for interacting with the automatic mesh refinement functionality that already exists in finite element libraries like LibMesh and Fenics. The overall goal is to create a generic VnV test that can be attached to the main function of the executable such that it automates the process of running mesh refinement and mesh convergence studies. In the case of UQ and SA, the project team will develop an interface for specifying the tools available in DAKOTA as VnV tests. }

{ RNET will be responsible for this task with ORNL providing input and guidance. }

2.3.5 Task 5: Integration into the NEAMS tool-chain

{ In this task, the program team will integrate the VnV framework into a variety of real applications. Initially, this testing will be completed in tools used heavily in the NEAMS toolkit; MOOSE, libMesh and PETSc, but other third party applications will also be investigated. To ensure seamless integration with the MOOSE tools, the project team will re-implement the XML based configuration file using the MOOSE input file format. This will allow for the configuration of the VnV tests in a MOOSE application directly from the input file.

The goal of this task will be to generate informative, production quality VnV reports for a number of examples available in the MOOSE testing suite. Doing so allows us to test every facet of the proposed framework, while also acting as the first demonstration of the value provided by the framework. These results of these tests will be hosted on the RNET website as they become available. }

{ RNET will be responsible for this task and ORNL will provide guidance on various technical implementations and details. }

2.3.6 Task 6: Development of an interface for the NEAMS workbench

{ In this task, the project team will integrate the toolkit directly into the NEAMS workbench. This will be a two stage process. First, the project team will implement the required interface files for enabling the context aware auto-complete features available in the NEAMS workbench. MOOSE based input files are already largely supported in the workbench; however, there will likely be some issues with determining which tests are applicable at which injection points. Second, we will create the interfaces for viewing the HTML based VnV reports inside the QT based workbench. As demonstrated in Phase I, this will be completed using the QWebEngineView components support for displaying HTML files inside QT windows. }

{ RNET will be responsible for this task }

2.3.7 Task 7: Preparation for the First Release

{In this task, the project team will prepare the package for its first release. This will include writing documentation ,user manuals and tutorials.

An important goal of this task will be to develop a mechanism for reducing the size of the VnV reports. In particular, this task will involve modifying the report generation system such that it maximizes data reuse in situations where an injection point or test is encountered multiple times within a simulation. Support for lazy-loading of content in collapsed sections will also be implemented. Such features are not critical to the functionality of the toolkit; however, they will significantly improve the usability of overall product. The custom markdown format will also be hardened, extended and released as part of this task. }

{ RNET will work on the implementation for this task }

2.4 Facilities/Equipment

2.4.1 RNET Facilities

RNET has the necessary office equipment to manage an SBIR/STTR contract including networks, workstations, and accounting software. In addition, RNET has the tools (software and hardware) to evaluate and develop the technologies proposed here.

RNET currently has 9 development computers and a 10-node development cluster that can be used for development and testing in this effort. Each cluster node has two quad-core or hexa-core XEON CPUs, 24-32GB of DRAM, 500+GB of local disk. Two data networks are available, a COTS 1 Gbps Ethernet network and a 10 Gbps Ethernet network. The Linux development nodes and the RNET cluster have the necessary Linux/GNU toolchains and development environments including; GNU tool chain, Microsoft .Net Framework, and Java Standard Edition.

2.4.2 ORNL Facilities

The Oak Ridge National Laboratory (ORNL) hosts three petascale computing facilities: the Oak Ridge Leadership Computing Facility (OLCF), managed for DOE; the National Institute for Computational Sciences (NICS) computing facility operated for the National Science Foundation (NSF); and the National Climate-Computing Research Center (NCRC), formed as collaboration between ORNL and the National Oceanographic and Atmospheric Administration (NOAA) to explore a variety of research topics in climate sciences. Each of these facilities has a professional, experienced operational and engineering staff comprising groups in high-performance computing (HPC) operations, technology integration, user services, scientific computing, and application performance tools.

ORNL also has the Compute and Data Environment for Science (CADES) which is a fully integrated infrastructure offering compute and data services for researchers lab-wide. We will work with appropriate program managers to apply for allocation requests as appropriate.

The ORNL computer facility staff provides continuous operation of the centers and immediate problem resolution. On evenings and weekends, operators provide first-line problem resolution for users with additional user support and system administrators on-call for more difficult problems. ORNL also has state-of-the-art visualization facilities that can be used on site or accessed remotely.

3 Consultants and Subcontractors

Oak Ridge National Laboratory(ORNL) is the Research Institution for this proposal and will serve as a subcontractor for this SBIR.

3.1 Gregory Watson

Gregory Watson, PhD, is a Senior Research Scientist in the Computer Science Research Group at Oak Ridge National Laboratory. Dr. Watson's research interests include programming tools and development environments for high performance and scientific computing, software engineering practices, reproducibility, and education and training for scientists. Dr. Watson is the founder of the Eclipse Parallel Tools Platform, a project that was originally started as a collaboration between Los Alamos National Laboratory and IBM in 2004, and that continues to be used across laboratories, academia, and industry. He is also a founding member of the Eclipse Science Working Group, and project leader of the Eclipse Science Top Level Project. Dr. Watson has considerable experience developing and implementing efficient parallel debugging software for high performance computing environments and has a wealth of experience in the development of highly scalable tools and communication frameworks for peta-scale high performance computing systems.

4 Principal Investigator and other Key Personnel

4.1 Ben O'Neill, PI

Ben O'Neill is a Research Scientist at RNET and will be the PI on this project. Ben is a full time employee at RNET and has sufficient time to dedicate to this project. Ben is a permanent resident in the USA (citizen of New Zealand). The proposed work does not include any Export Control restriction, as such, this work visa should be sufficient. In addition to the current project, Ben is the lead developer in RNET's Cloudbench project which aims to develop a web-enabled interface for remote execution and visualization for nuclear physics tools. Ben was also heavily involved in the development of the SolverSelector framework for facilitating automatic linear solver selection in high performance applications through machine learning. His background is in Applied mathematics with a focus on high performance computing and parallel-time integration. His thesis work involved the optimization and implementation of a parallel time integration codes for nonlinear PDEs including implementing a fully adaptive and parallel space-time solver using the Fenics finite element package.

4.2 Gerald Sabin

Dr. Gerald Sabin, Project Manager at RNET, will be the senior advisor for this project. Dr. Sabin is a full time employee of RNET, and has sufficient time to dedicate to project tasks as indicated in the cost proposal. Since he is a US Citizen, he can undertake relevant integration work in Export-Controlled areas of the project, if necessary. Currently, he is working on several Scientific Computing (HPC) SBIR/STTR projects at RNET. He is the PI for this Phase II SBIR Cloudbench project (DE-SC0015748) and the ongoing Phase II DOE SBIR for the Automated Solver Selection for Nuclear Engineering Simulations. He has also worked on distributed memory, GPU, multi-core and SIMD optimizations to the Air Force's Kestrel code (DOD Contract#:FA9550-12-C-0028) and has also been involved in developing fine-grained power profiling hardware and software tools for HPC application profiling (DOE Contract#:DE-SC0004510). He has also been the PI on several other related projects including a NASA Phase I project developing SIMD optimizations for Monte Carlo codes (NNX14CA44P), developing parallelization optimizations for PETSc (DOE Contract#: DE-SC0002434), developing data virtualization support and bitmap indexing for massive Climate Modeling data sets (DOE Contract #:DE-SC0009520), and developing the SmartNIC software stack for application-aware network offloading (DOE Contract#: DE-FG02-08ER86360).

5 Related Work

RNET and ORNL have past and current experience in several SBIR/STTR projects on modeling and simulation, high performance computing, and large data formats. Some of these projects are briefly described below.

5.1 RNET's Related Work

5.1.1 Automated Solver Selection for NEAMS Tools

RNET in collaboration with University of Oregon (Prof. Boyana Norris) is developing an add-in feature for the NEAMS toolkit being developed by the Department of Energy. This work is being done as part of DOE Phase II STTR project (Contract Number DE-SC0013869). This add-in feature being developed by RNET leverages machine learning techniques to automatically select the optimal solver based on run-time dependent features of the problem and the underlying compute architecture with minimal runtime overhead in solver selection during the course of NEAMS simulations.

5.1.2 Cloud-based Scientific Workbench for Nuclear Reactor Simulation Life Cycle Management

The predictive modeling approaches and softwares being continually developed and updated by the DOE nuclear engineering scientists (under programs such as NEAMS, CASL, RISMC etc.) need to be efficiently transferred to the nuclear science and engineering community. An advanced workflow management workbench is required to allow efficient usage from small and large business and research groups. The workbench must manage inputs decks, simulation execution (on a local machine, a High Performance Compute cluster, or a Cloud cluster), intermediate results, final results and visualizations, and provenance of the tools and settings. CloudBench is a hosted simulation environment for large scale numeric simulations. CloudBench will augment existing simulation, Integrated Development Environment, and workbench tools being developed by the DOE and industry. It offers a complete set of simulation management features not available in open tools: sharing of configurations, simulation output, and provenance on a per simulation or per project basis; multi-simulation provenance history to allow simulations to be reconstructed, verified, or extended; and remote access to simulation tools installed on Cloud and HPC resources. The portal enables easy adoption of government codes.

5.1.3 Scaling the PETSc Numerical Library to Petascale Architectures

RNET has developed an extended version of the numerical library PETSc [8] in collaboration with Ohio State University and Argonne National Lab. PETSc is an MPI-based numerical library of linear and nonlinear solvers that is widely used in a variety of scientific domains. With the emergence of multicore processors and heterogeneous accelerators as the building blocks of parallel systems, it is essential to restructure the PETSc code to effectively exploit multi-level parallelism. Changes to the underlying PETSc data structures are required to leverage the multicore nodes and GPGPUs being added to the “cluster architectures”.

This project was funded by Department of Energy under the STTR program from August 2010 (Contract Number DE-SC0002434) to May 2013. Dr. P. Sadayappan (OSU) and Dr. Boyana Norris (ANL) have played a key role in this effort by serving as technical advisors. As part of the project, the team has investigated ways for the PETSc library to fully utilize the computing power of future Petascale computers. Novel sparse matrix types, vector types, and preconditioning techniques that are conducive for GPU processing and SIMD parallelization have been integrated into the PETSc library. The matrix vector operations have been optimized for specific architectures and GPUs by utilizing the autotuning tools.

5.2 ORNL’s Related Work

Dr. Watson is the main developer for the Eclipse Parallel Tools Platform (PTP). The aim of the PTP project is to produce an open-source industry-strength platform that provides a highly integrated environment specifically designed for parallel application development. The platform provides a standard, portable parallel IDE that supports a wide range of parallel architectures and runtime systems; a scalable parallel debugger; support for the integration of a wide range of parallel tools; and an environment that simplifies the end-user interaction with parallel systems.

In addition, Dr. Watson has played a significant role in the development of GAURD, a parallel relative debugger for high performance systems. Unlike other conventional parallel debuggers ,a relative debugger provides the ability to dynamically compare data between two executing programs regardless of their location and configuration. In GAURD, data comparisons can be performed either using an imperative scheme or a declarative scheme. Imperative comparisons can be performed explicitly by the user when two programs under the control of the debugger are stopped at breakpoints.

A solid foundation has been laid to achieve the objectives of the proposed project. The final piece of the puzzle is the optimization and hardening of these core approaches for efficient execution in real applications. Once this is accomplished, it will be possible to seamlessly integrate functionality for end-user V&V into any general purpose numerical simulation software. The collaboration between ORNL and RNET is a perfect fit to facilitate this final step. RNET brings to the table the required computer science expertise to satisfy the needs of this project, as is evident from a description of their related work, and ORNL brings a wealth of experience developing novel HPC software solutions, developing robust debugging tools for use in HPC environments and working with a range of real application codes.

References

[1] DAKOTA - Explore and Predict with Confidence . https://dakota.sandia.gov/.

[2] DoD VV&A Recommended Practices Guide . https://vva.msco.mil/.

[3] NEAMS Software Verification and Validation Plan Requirements V0. https://preview.tinyurl.com/y8q7zr2v.

[4] Neams: The nuclear energy advanced modeling and simulation program. http://people.nas.nasa.gov/~pulliam/Overflow/Overflow_Manuals.html.

[5] Allison H Baker, Robert D Falgout, and Ulrike Meier Yang. An assumed partition algorithm for determining processor inter-communication. *Parallel Computing*, 32(5-6):394–414, 2006.

[6] MR Collins, Franck J Vecchio, Robert G Selby, and Pawan R Gupta. The failure of an offshore platform. *CONCRETE INTERNATIONAL-DETROIT-*, 19:28–36, 1997.

[7] Bernt Jakobsen and Finn Rosendahl. The sleipner platform accident. *Structural Engineering International*, 4(3):190–193, 1994.

[8] D. Lowell, J. Holewinski J. Godwin, D. Karthik, C. Choudary, A. Mametjanov, B. Norris, G. Sabin, P. Sadayappan, and J. Sarich. Stencil-aware gpu optimization of iterative solvers. *SIAM Journal on Scientific Computing*, 35(5), 2013.

[9] A Pinar and B Hendrickson. Communication support for adaptive communication. *In Proceedings of the 10th SIAM Conference on Parallel Processing*, 2001.

Education

- University of Colorado at Boulder, Boulder, CO
  - Ph.D. in Applied Mathematics, August 2017.
  - Masters in Applied Mathematics, May 2015.
- The University of Waikato, Waikato, NZ
  - Bachelor of Science (Hons) in Mathematics and Physics, December 2011.

Selected Research & Professional Experiences

- Senior Researcher, RNET Technologies, July 2017 - Present
  - An Extensible Verification and Validation Library with NEAMS Workbench Integration. DOE Phase I SBIR, Primary Investigator (DE-SC0018728) July 2018 - Present
  - VERA-Workbench: A unified Multi-physics toolkit for the VERA Suite of Tools. DOE Phase I SBIR, Primary Investigator (DE-SC0017701) October 2017 - May 2018
  - Cloud-based Scientific Workbench for Nuclear Reactor Simulation Life Cycle Management. DOE Phase II SBIR, Research Scientist (DE-SC0015748) October 2017 - Present
  - Automatic solver selection for Nuclear Engineering Simulation. DOE Phase II SBIR, Research Scientist (DE-SC0013869) July 2017 - Present
- Graduate Student, University of Colorado at Boulder, April 2014-August 2017
  Mentor: Tom Manteuffel (CU Boulder), Jacob Schroder (Lawrence Livermore National Laboratory)
  - Studied Parallel-in-time solvers for parabolic partial differential equations
  - Implemented a fully adaptive parallel-in-time solver using the FENICS finite element package, C/C++, and MPI.
  - Designed and implemented a temporal load balancing algorithm for the opensource XBraid project with O(log(P)) communication.
- Summer Internship, May-August , 2014-2016 Lawrence Livermore National Laboratory
  Mentor: Jacob Schroder and Rob Falgout
  - Researched an embedded error estimate for the XBraid project.
  - Designed and implemented a cost efficient parallel-in-time solver based on MGRIT and Richardson Extrapolation.

Refereed Publications

- R. D. Falgout, T. A. Manteuffel, B. O'Neill, and J. B. Schroder, Multigrid reduction in time for nonlinear parabolic problems, Copper Mountain Special Section, SIAM J.Sci. Comput. (accepted), (2016). LLNL-JRNL-692258 .
- M.T. Wilson, P.A. Robinson, B. O'Neill., D.A. Steyn-Ross, Complementarity of Spike- and and Rate-Based Dynamics of Neural Systems, PLOS Computational Biology, Vol 8 (2012).

Education

- Doctor of Philosophy, Computer Science and Engineering, The Ohio State University, Columbus, OH, 2006. Advisor: Dr. P. Sadayappan. Doctoral Thesis: **Fairness in Parallel Job Scheduling.**
- Master of Science, Computer Science and Engineering, The Ohio State University, Columbus, OH, 2004. Advisor: Dr. P. Sadayappan.
- Bachelor of Science, Magna Cum Laude, Computer Science and Engineering Physics with specializations in Computer Engineering and Electrical Engineering, John Carroll University, University Heights, OH, 2002

Selected Research & Professional Experiences (from over 25 RNET Projects)

- Project Manager/Senior Researcher, RNET Technologies, July 2005 - Present

  2019 DOE SBIR: Tensor Contraction and Operation Minimization for Extreme Scale Computational Chemistry (Phase I, Principal Investigator)

  2016 DOE SBIR: A Self-Configuring Machine Learning Tool for Predicting Optimal Numerical Methods (Phase II, Principal Investigator, DE-SC0013869)

  2016 DOE SBIR: Web Infrastructure for Remote Modeling and Simulation of Nuclear Reactors and Fuel Cycle Systems (Phase II, Principal Investigator, DE-SC0015748)

  2015 DOE SBIR: iNFORMER: A MapReduce-like Data-Intensive Processing Framework for Native Data Storage and Formats (Phase II, Principal Investigator, DE-SC0011312)

  2015 DARPA SBIR: Performance Portable Framework for Developing Graph Applications (Phase II, Principal Investigator, D16PC00183)

  2015 DARPA SBIR: Ultra-High Productivity Graph Processing using NVidia Optimized Giraph (Phase I, Principal Investigator, W911NF-15-P-0043)

  2014 NASA SBIR: Domain Specific Language based Parallelization for Geant4 for Space-based Applications on Current and Future Architectures (Phase I, Principal Investigator, NNX14CA44P)

  2010 DOE STTR: HPC Application Energy Measurement and Optimization Tools (Phase II, Principal Investigator, DE-SC0004510)

  2010 Air Force STTR: Scalable Multi-Tiered CFD and CSD Codes for Kestrel (Phase II, Principal Investigator, FA9550-12-C-0028)

  2009 DOE SBIR:Enhancement of GridFTP Performance Through GMPLS Integration and Hardware Offloading (Phase II, Principal Investigator, DE-SC0002182)

  2009 DOE SBIR:Optimization of the PETSc Library for Clusters of MultiCore Processors (Phase II, Principal Investigator, DE-SC0002434)

  DOE43d: FPGA-based Multiprocessor Architecture for LQCD Computations SBIR Phase I 2006 (Phase I, Senior Researcher, DE-FG02-05ER84164)

Selected Refereed Publications (from over 20 publications)

- “Long-haul Secure Data Transfer using Hardware-assisted GridFTP,” Mohammad Rashti, Gerald Sabin, Rajkumar Kettimuthu, in FUTURE GENER COMP SY Journal, Elsevier Science, accepted for publication.
- “Automatic Kernel Acceleration of PETSc Krylov Solvers”, Chekuri Choudary, Deepan Balasubramanian, Jeswin Godwin, Daniel Lowell, Azamat Mametjanov Boyana Norris, P. Sadayappan, Gerald Sabin, Sravya Tirukkovalur, in TWELFTH COPPER MOUNTAIN CONFERENCE ON ITERATIVE METHODS 2012
- “Enhancement of GridFTP through Hardware Offloading”, M. Rashit, R. Kettimiuthu, G. Sabin, SC14 SCinet Workshop: Innovating the Network for Data Intensive Science
- “Stencil-Aware GPU Optimization of Iterative Solvers,C. Choudary, J. Godwin, J. Holewinski, D. Karthik, D. Lowell, A. Mametjanov, B. Norris, G. Sabin, P. Sadayappan, SISC, Copper Mountain Special Section 2012. Preprint ANL/MCS-P3008-0712, July 2012.
- “Scaling a Numerical Library to Emerging Compute Architectures”, Chekuri S. Choudary, Jeswin Godwin, Deepan Karthik, Daniel Lowell, Boyana Norris, Gerald Sabin, P. Sadayappan, Sravya Tirukkovalur in ”SIAM Conference on Parallel Processing for Scientific Computing” 2012
- An Integrated Approach to Locality Conscious Processor Allocation and Scheduling of Mixed Parallel Applications,N. Vydyanathan, S. Krishnamoorthy, G. Sabin, U. Catalyurek, T. Kurc, P. Sadayappan, and J. Saltz, IEEE Transaction on Parallel and Distributed Systems, Vol. 20, No. 9, Aug 2009
- A message passing benchmark for unbalanced applications,James Dinan, Stephen Olivier, Gerald Sabin, Jan Prins, P. Sadayappan, Chau-Wen Tseng, Journal of Simulation Modeling Practice and Theory (SIMPAT). Volume 16, Issue 9, Pages 1177-1189. October, 2008.
- An Integrated Approach for Processor Allocation and Scheduling of Mixed-Parallel Applications, Nagavijayalakshmi Vydyanathan, Sriram Krishnamoorthy, Gerald Sabin, Umit Catalyurek, Tahsin Kurc, P. Sadayappan, and Joel Saltz, Internal Conference on Parallel Processing 2006.
- Assessment and Enhancement Of Meta-Schedulers for Multi-Site Job Sharing, Gerald Sabin, Vishvesh Sahasrabudhe, P. Sadayappan, International Symposium on High Performance Distributed Computing 2005.
- On Fairness in Distributed Job Scheduling Across Multiple Sites, Gerald Sabin, Vishvesh Sahasrabudhe, P. Sadayappan, Cluster 2004.
- Job Fairness in Non-Preemptive Job Scheduling, Gerald Sabin, Garima Kochhar, P. Sadayappan, International Conference on Parallel Processing 2004.
- Scheduling of Parallel Jobs on a Heterogeneous Cluster of Multi-Processor Systems, Gerald Sabin, Rajkumar Kettimuttu, Arun Rajan, P. Sadayappan, Workshop On Job Scheduling Strategies For Parallel Processing 2003.

Education

- Wright State Univeristy (WSU), Dayton, OH
  - Ph.D. in Computational Science and Mathematics, August 2019.
- California State University Northridge (CSUN), Northridge, CA
  - M.S. in Applied Mathematics, August 2015.
- San Jose State Univeristy (SJSU), San Jose, CA
  - Bachelor of Art in Mathematics, May 2011.

Experiences

- Researcher, RNET Technologies, April 2019 - Present
  - Tensor Contraction and Operation Minimization for Extreme Scale Computational Chemistry.
- Computational Mechanics Intern, Applied Optimization, May 2018 - March 2019
  - Development an in-house solver for the Nonlinear Thermal/Distortion Analysis of Materials.
- Graduate Research Assistant, CSUN-WSU, April 2014-Present
  Mentors: Alex Alekseenko (CSUN), Amit Sharma (WSU), Mohamed Sulman (WSU), Aihua Wood (AFIT)
  - Developing a fast deterministic-stochastic method for computing the Boltzmanns collision operator in $O(MN)$ operations where N is the degree of freedom in the three dimensional velocity space and $M < N$ using a machine learning approach to estimate the Boltzmann's velocity distribution function as a Gaussian Mixtures Model.
  - Developing an efficient solver for numerical modeling of solutions to a system of nonlinear equations that capture the collapse phenomenon of plasmas density in the Ionosphere layer of the Earth.
  - Developing a positivity preserving adaptive moving mesh finite element methods for nonlinear diffusion-reaction-chemotaxis systems of PDEs.

Publications

- M. Sulman, T. Nguyen, A Positivity Preserving Moving Mesh Method for Cancer Invasion Model, J. Computational and Applied Math, (in review).
- M. Sulman, T. Nguyen, A Positivity Preserving Moving Mesh Finite Element Method for the Keller-Segel Chemotaxis Model, J. Sci. Compt., (2019).
- A. Alekseenko, T. Nguyen, A. Wood, A deterministic-stochastic method for computing the Boltzmann collision integral in $O(MN)$ operations, Kinetic & Related Models, Vol 11, 2018.

GREGORY R. WATSON

CURRICULUM VITAE

EDUCATION

2000	Doctor of Philosophy, School of Computer Science and Software Engineering, Monash University
1985	Bachelor of Science with First Class Honours (Information Science), Department of Computer Science, University of Tasmania
1984	Bachelor of Science, Department of Computer Science, University of Tasmania

SUMMARY OF WORK EXPERIENCE

2016-	Senior Research Scientist, Computer Science Research Group, Oak Ridge National Laboratory
2014-2017	Adjunct Professor, NYU Center for Data Science
2010-2016	HPC Tools Architect, Cross Platform Software, System & Technology Group, IBM Corporation
2007-2010	Project Leader, Eclipse Parallel Tools Platform, High Productivity Computing Group, T.J. Watson Research Center, IBM Corporation
2006-2007	Team Leader, Cluster Research Team, Advanced Computing Laboratory, Los Alamos National Laboratory
2002-2006	Technical Staff Member, Advanced Computing Laboratory, Los Alamos National Laboratory
2000-2002	Senior Research Fellow, School of Computer Science and Software Engineering, Monash University
1997-2000	Research Fellow, School of Computer Science and Software Engineering, Monash University
1994-1997	General Manager, Queensland Parallel Supercomputing Facility
1991-1994	Manager, Corporate Computer Systems, Division of Information Services, Griffith University
1988-1991	UNIX and Networking Consultant, Information Technology Centre, Griffith University
1986-1988	Research Programmer, Department of Computer Science, University of Tasmania

AWARDS

- R&D 100 award for the "Clustermatic" project to develop the world's first 2000+ CPU cluster system (2004)
- Los Alamos Achievement Award (2003, 2004, 2006)
- ClusterWorld Excellence in Clustering Technology (2004)
- Defence Program Award of Excellence (2002, 2003)
- Prime Computer prize for the greatest proficiency in honors-level study in the subject of Information Science (1986)

RESEARCH CONTRACTS

- NFS award #1047956, October 2010 – September 2013, "SI2-SSI: A Productive and Accessible Development Workbench for HPC Applications Using the Eclipse Parallel Tools Platform", PI: Jay Alameda (NCSA), co-PI: Steven Brandt (LSU), co-PI: Allen D Maloney (UO), co-PI: Marc Snir (Illinois), co-PI: Gregory Watson (IBM). Amount awarded: $1,434,000
- DOE Office of Science award ER25928, January 2009 – December 2011, "A Scalable Development Environment for Peta-Scale Computing", PI: Gregory Watson, co-PI: Jeremy C. Smith (ORNL), co-PI: Randy Roberts (LANL), co-PI: Wolfgang Frings (JSC), co-PI: David Abramson (Monash). Amount awarded: $1,152,000

ACHIEVEMENTS AND COMMUNITY CONTRIBUTIONS

- Project Leader of the Eclipse Science Top Level Project (http://science.eclipse.org)
- Project Leader of the Eclipse Parallel Tools Platform project (http://eclipse.org/ptp)
- Member of the Eclipse Tools Project Management Committee (http://projects.eclipse.org/projects/tools/developers)
- Founding member of the Scalable Tools Communication Infrastructure project (http://stci.wikidot.com)
- Contributor of the PowerPC port to the Coreboot project (previously known as LinuxBIOS) (http://www.coreboot.org)
- Contributor to the v9fs project (http://v9fs.sourceforge.net)
- Past-Chair of the auDA Foundation (http://www.audafoundation.org.au)
- Founding board member and past-Chair of the auDA, the Australia Domain Name Administration (http://www.auda.org.au/board/members)
- Past-President and Director of the Internet Society of Australia (http://www.isoc-au.org.au/Organisation/PastDirs.html)

Title	Topic Number	Source	Location	Amount (Fee Included)	Start	End	Status	Total Months Allocated				
								Gerald Sabin	Ben Oneill	Truong Nguyen	Jamar Robinson	Pavan Yallamelli
An Extensible Verification and Validation Library with NEAMS Workbench Integration	DOE 18.2-30d	DOE Phase I	Dayton, OH	$150,000	7/2/18	5/1/19	Current	2	7			
Tensor Contraction and Operation Minimization for Extreme Scale Computational Chemistry	DOE 19.1-07b	DOE Phase I	Dayton, OH	$225,000	2/19/19	2/18/20	Current	3		6		
Performance Portable Framework for Developing Graph Applications	SB151-004	DARPA Phase II	Dayton, OH	$1,000,000	6/1/16	5/30/19	Current	16	4			
Automated Solver Selection for Nuclear Engineering Simulations	DOE15.2-32d	DOE Phase II	Dayton, OH	$1,000,000	8/1/16	7/31/19	Current	6	10		3	
Web Infrastructure for Remote Modeling and Simulation of Nuclear Reactors and Fuel Cycle Systems	DOE16.2-30d	DOE Phase II	Dayton, OH	$1,000,000	7/31/17	7/30/19	Current	2.5	12			14

Ben ONeill's Monthly Commitment																									
Topic	8-19	9-19	10-19	11-19	12-19	1-20	2-20	3-20	4-20	5-20	6-20	7-20	8-20	9-20	10-20	11-20	12-20	1-21	2-21	3-21	4-21	5-21	6-21	7-21	8-21
DOE 18.2-30d	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SB151-004	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DOE15.2-32d	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DOE16.2-30d	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Monthly Total	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Title	Topic Number	Source	Location	Amount (Fee Included)	Start	End	Status	Total Months Allocated				
								Gerald Sabin	Ben Oneill	Truong Nguyen	Jamar Robinson	Pavan Yallamelli
An Extensible Verification and Validation Library with NEAMS Workbench Integration	DOE 18.2-30d	DOE Phase I	Dayton, OH	$150,000	7/2/18	5/1/19	Current	2	7			
Tensor Contraction and Operation Minimization for Extreme Scale Computational Chemistry	DOE 19.1-07b	DOE Phase I	Dayton, OH	$225,000	2/19/19	2/18/20	Current	3		6		
Performance Portable Framework for Developing Graph Applications	SB151-004	DARPA Phase II	Dayton, OH	$1,000,000	6/1/16	5/30/19	Current	16	4			
Automated Solver Selection for Nuclear Engineering Simulations	DOE15.2-32d	DOE Phase II	Dayton, OH	$1,000,000	8/1/16	7/31/19	Current	6	10		3	
Web Infrastructure for Remote Modeling and Simulation of Nuclear Reactors and Fuel Cycle Systems	DOE16.2-30d	DOE Phase II	Dayton, OH	$1,000,000	7/31/17	7/30/19	Current	2.5	12			14

Gerald Sabin's Monthly Commitment																									
Topic	8-19	9-19	10-19	11-19	12-19	1-20	2-20	3-20	4-20	5-20	6-20	7-20	8-20	9-20	10-20	11-20	12-20	1-21	2-21	3-21	4-21	5-21	6-21	7-21	8-21
DOE 18.2-30d	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DOE 19.1-07b	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SB151-004	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DOE15.2-32d	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DOE16.2-30d	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Monthly Total	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Title	Topic Number	Source	Location	Amount (Fee Included)	Start	End	Status	Total Months Allocated				
								Gerald Sabin	Ben Oneill	Truong Nguyen	Jamar Robinson	Pavan Yallamelli
An Extensible Verification and Validation Library with NEAMS Workbench Integration	DOE 18.2-30d	DOE Phase I	Dayton, OH	$150,000	7/2/18	5/1/19	Current	2	7			
Tensor Contraction and Operation Minimization for Extreme Scale Computational Chemistry	DOE 19.1-07b	DOE Phase I	Dayton, OH	$225,000	2/19/19	2/18/20	Current	3		6		
Performance Portable Framework for Developing Graph Applications	SB151-004	DARPA Phase II	Dayton, OH	$1,000,000	6/1/16	5/30/19	Current	16	4			
Automated Solver Selection for Nuclear Engineering Simulations	DOE15.2-32d	DOE Phase II	Dayton, OH	$1,000,000	8/1/16	7/31/19	Current	6	10		3	
Web Infrastructure for Remote Modeling and Simulation of Nuclear Reactors and Fuel Cycle Systems	DOE16.2-30d	DOE Phase II	Dayton, OH	$1,000,000	7/31/17	7/30/19	Current	2.5	12			14

Truong Nyugen Monthly Commitment																										
Topic	Monthly Fraction	8-19	9-19	10-19	11-19	12-19	1-20	2-20	3-20	4-20	5-20	6-20	7-20	8-20	9-20	10-20	11-20	12-20	1-21	2-21	3-21	4-21	5-21	6-21	7-21	8-21
DOE 18.2-30d	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DOE 19.1-07b	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SB151-004	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DOE15.2-32d	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DOE16.2-30d	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Monthly Total		0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

1 Section A – Senior/Key Personnel

Dr. Ben O'Neill is the PI and will lead the technical effort with assistance from Dr. Gerald Sabin and Truong Nguyen. Dr. Sabin will also assist Dr. O'Neill with the management aspects of the Phase II project. Rates and hours are included in the cost proposal.

2 Section B – Other Personnel

Pavan Yallamelli, Research Scientist, will assist with implementation and testing.

3 Section C – Equipment Description (Items Exceeding $5,000)

No major equipment.

4 Section D – Travel

There will be two meetings with the DOE Program Manager and interested DOE scientists. The first will be a kick-off meeting (i.e., the DOE PI Meeting) to start the contract and ensure that the DOE needs are met. The second is a review meeting toward the end of the project to discuss the project status. RNET is not budgeting any cost for travel. RNET travel expenses (e.g., if both meetings are in person) will be paid from internal funding.

5 Section E – Participant/Trainee Support Costs

None

6 Section F – Other Direct Costs

6.1 Subcontract

Oakridge National Laboratory is subcontractor on this project. The ORNL budget and a letter of commitment are attached to this proposal submission.

7 Section H – Indirect Costs

RNET uses a labor overhead of 51%. This overhead includes, fringe benefits (holidays, sick leave, vacation, health-care benefits and retirement contributions), payroll taxes, and other indirect charges.

RNET charges 7% for "General and Administrative" tasks (G&A), including accounting, payroll services, office rent, office supplies, telephone, Internet connection, etc.

8 Section J – Fee

RNET charges a fee (profit) of slightly less than 7%. This fee is consistent with other accepted RNET contracts.

OAK RIDGE
National Laboratory

PO Box 2008
Oak Ridge, TN 37831-6163
(865) 574-6224| nicholsja@ornl.gov

April 18, 2019

Dr. V. Nagarajan, President
RNET Technologies, Inc.
240 West Elmwood Drive
Dayton, Ohio 45459-4248

Dr. Nagarajan:

Letter of Support for ORNL Personnel to Participate in Project: VnV: A Self Documenting Testing Framework for In-situ Verification and Validation in High Performance Computing Applications

This is to confirm Oak Ridge National Laboratory research scientist, Dr. Gregory Watson, will participate in the proposed project, "VnV: A Self Documenting Testing Framework for In-situ Verification and Validation in High Performance Computing Applications," in response to DOE SBIR-STTR FY 2019 Phase II Release 2 Funding Opportunity Announcement DE-FOA 0001976.

Pending award and all required DOE approvals for this project, Dr. Watson will provide expertise and software development support for large scale distributed tests, a type safe injection point system using LLVM, and a lightweight distributed test execution framework. Dr. Watson will also provide expertise and software development support for integrating job parallelism features of the Eclipse Parallel Tools Platform.

Dr. Watson will commit 1,393 hours during the period of performance; budget table provided on page 2. If you need further information, please do not hesitate to contact me at 865-574-6224 or nicholsja@ornl.gov.

Sincerely,

Jeffery A. Nichols
Associate Laboratory Director
Computing and Computational Sciences

JAN:ltm

c: File–NoRC

VnV: A Self Documenting Testing Framework for In-situ Verification and Validation in High Performance Computing Applications," in response to DOE SBIR-STTR FY 2019 Phase II Release 2 Funding Opportunity Announcement DE-FOA 0001976.

Estimated Budget by Fiscal Year				
Category	**FY 2019**	**FY 2020**	**FY 2021**	**Total**
Labor	$14,099	$123,913	$109,592	$247,604
Travel	$243	$2,057	$1,812	$4,112
Overhead	$5,598	$49,186	$43,501	$98,285
Subtotal	$19,940	$175,156	$154,904	$350,000
Grand Total	**$19,940**	**$175,156**	**$154,904**	**$350,000**
LDRD	$798	$7,006	$6,196	$14,000

Department of Energy

ORNL Site Office
P.O. Box 2008
Oak Ridge, Tennessee 37831-6269

April 17, 2019

Mr. Carl Hebron
Program Coordinator
Small Business Innovation Research/
Small Business Technology Tranfer
Program Office
U.S. Department of Energy
1000 Independence Avenue, Southwest
Washington, DC 20585-1290

Dear Mr. Hebron:

DEPARTMENT OF ENERGY (DOE) FISCAL YEAR (FY) 2019 SMALL BUSINESS INNOVATION RESEARCH (SBIR)/SMALL BUSINESS TECHNOLOGY TRANSFER (STTR) PHASE II, RELEASE 2, FUNDING OPPORTUNITY ANNOUNCEMENT DE-FOA-0001976 (INTERNAL REFERENCE NUMBER NFE-19-07729)

UT-Battelle, LLC is the managing and operating contractor for the Oak Ridge National Laboratory (ORNL) under DOE Prime Contract Number DE-AC05-00OR22725. As a DOE Office of Science, Federally Funded Research and Development Center, ORNL is requesting authorization to team with RNET Technologies, Inc. in response to the FY 2019 Phase II, Release 2, of the DOE SBIR/STTR Solicitation DE-FOA-0001976. ORNL has submitted to the ORNL Site Office a statement of work (SOW) to assist in the development of a generic testing framework for designing and injecting self-documenting verification and validation software components into advanced numerical simulations. This SOW has been reviewed and approved according to our normal procedures and will be performed in accordance with contract number DE-AC05-00OR22725.

Authorization is granted for ORNL to participate in the proposed effort. The work proposed for the laboratory is consistent with, or complementary to, the missions of the laboratory, will not adversely impact execution of the DOE assigned programs at the laboratory, and will not place the laboratory in direct competition with the domestic private sector.

If there are any questions or additional information is required, please contact me at (865) 576-9262 or David Myers at (865) 576-5629.

Sincerely,

KENNETH KIMBROUGH
Digitally signed by KENNETH KIMBROUGH
Date: 2019.04.17 10:57:34 -04'00'

Kenneth L. Kimbrough
Contracting Officer

ORNL BUDGET JUSTIFICATION

Cost estimates presented in this proposal have been reclassified to be comparable with proposals from other research institutions. At Oak Ridge National Laboratory (ORNL), actual costs will be collected and reported in accordance with the Department of Energy (DOE) approved cost accounting system. Total cost presented in this proposal and actual costs will be equivalent, as will the subtotal of direct and indirect costs. Details of the budget breakdown of all budget categories, including salaries, benefits and fringe, and directs costs are described below. In consideration of cost estimates for ORNL staff support, it is important to understand that all ORNL costs are covered by proposals such as this one. There is no "base" funding for DOE Office of Science multiprogram national laboratories, including ORNL.

A. Senior/Key Personnel

ORNL's cost accounting system utilizes wage pools, which are calculated hourly rates for staff in similar job categories and salary ranges. Due to the use of these wage pools, the salary figure listed for Senior/Key Personnel represents the average salary for a staff scientist within a specific wage pool and not actual salary. For budgeting purposes, one calendar month is assumed to be 152 hours for ORNL staff. Fringe benefits are included in the ORNL employee wage pool rate at 39.6% for FY19 and 43.9% for out years for ORNL staff. Below is a summary of the role of each Senior/Key Person.

Gregory R. Watson, Principal Investigator

Calendar Months: 24

Dr. Watson will serve as Principal Investigator and will provide technical advice, system design, and development support for the project.

B. Other Personnel

Not applicable.

C. Equipment

Not applicable.

D. Travel

Travel amount includes estimated travel expense plus 2.8% travel administration value added tax (VAT) or "puddle" where G&A is applied on the puddle cost, but not the travel expense. Base travel cost is therefore made up of both the travel expense and 2.8% VAT.

Travel costs of $2,000/year (plus the 2.8% VAT) Years 1-2 be for 2 trips/year * 1 people @ ~$1000/trip at which the PI will present their research results and coordinate the project via a face-to-face meeting.

E. Participant/Trainee Support Costs

Not applicable.

F. Other Direct Costs

Line 8. Organizational Burden Administration

Organizational Burden Administration costs include utilities (purchased utilities as well as laboratory staff associated with maintaining the utility systems), space charges (building maintenance), division managerial oversight, technical and administrative support, and other support personnel such as plant and equipment, instrumentation and controls, environmental, safety, and health, finance and budget, quality, and health physics provided for the general benefit of all staff and R&D activities. Inclusion of these costs is necessary to provide a full accounting of estimated cost for the project period. All cost will be collected and reported in ORNL's cost accounting system, as approved by DOE.

G. Direct Costs

Total direct costs are $251,715 over project period.

H. Indirect Costs = $98,285 over project period.

ORNL overhead is applied on a cost element basis. ORNL overhead supports central management and administrative costs for the total laboratory, central maintenance activities, limited general plant equipment purchases (major items not covered by normal maintenance support), etc. For this proposal period, the composite ORNL Overhead rates are 39.46% for staff labor and 17.06% for travel.

ORNL Operating Rates

Description	**FY 2019**	**FY 2020**	**FY 2021**
General & Administrative	14.4%	14.4%	14.4%
Institutional	4.4%	4.4%	4.4%
LDRD	4.0%	4.0%	4.0%
Management Fee	0.9%	0.9%	0.9%
Travel Administration VAT	2.8%	2.8%	2.8%
Site Services	10.9%	10.9%	10.9%
SPP Safeguards & Security	3.5%	3.5%	3.5%
Composite Overhead Rate on:			
Labor	*39.46%*	*39.46%*	*39.46%*
Travel	*17.06%*	*17.06%*	*17.06%*
Fringe Benefits (ORNL Staff)	*39.6%*	*43.9%*	*43.9%*
Wage Pools ($/hour)	**FY 2019**	**FY 2020**	**FY 2021**
S&T D	$145.37	$152.37	$159.20
Organizational Burden ($/hour)	**FY 2019**	**FY 2020**	**FY 2021**
R&D Staff	$53.68	$54.75	$55.85

Full Cost Recovery Letter Page 1

Department of Energy
ORNL Site Office
P.O. Box 2008
Oak Ridge, Tennessee 37831-6269

January 11, 2018

Mr. J. Scott Branham
Chief Financial Officer
Oak Ridge National Laboratory
UT-Battelle, LLC
Post Office Box 2008
Oak Ridge, Tennessee 37831-6231

Dear Mr. Branham:

CONTRACT NUMBER DE-AC05-00OR22725, GENERIC LETTER ON UT-BATTELLE, LLC (UT-BATTELLE) OAK RIDGE NATIONAL LABORATORY RATES

The enclosed letter is furnished annually to answer questions on rates charged by UT-Battelle when performing work for sponsors other than the Department of Energy.

If there are any questions or additional information is required, please contact Rakisha Beatty at (865) 576-4522.

Sincerely,

Judson A. Kingman
Contracting Officer

Enclosure

cc w/enclosure:
Deborah U. Mann, ORNL
Jeremy D. Bivins, SC-OR
Jerry M. McKeehan, SC-OR
Michael D. Ashworth, SC-OSO
Rakisha M. Beatty, SC-OSO
Michele G. Branton, SC-OSO
H. Randall Fair, SC-OSO
Johnny O. Moore, SC-OSO

Full Cost Recovery Letter Page 2

Department of Energy

ORNL Site Office
P.O. Box 2008
Oak Ridge, Tennessee 37831-6269

January 11, 2018

To Whom It May Concern:

OAK RIDGE NATIONAL LABORATORY (ORNL) - FULL COST RECOVERY

UT-Battelle, LLC (UT-Battelle) manages and operates ORNL for the Department of Energy (DOE) under management and operating contract number DE-AC05-00OR22725. ORNL is a multi-program laboratory facility performing applied and basic scientific research. In addition to performing research for DOE, ORNL conducts research for the private sector, other Federal agencies, and states. All work is performed on the basis of full-cost recovery, as mandated by law. For ORNL, full cost includes all direct costs incurred in performing research work, all allocable costs incurred by DOE or UT-Battelle, and a Federal Administrative Charge (FAC) of three percent of these costs. The FAC does not apply when prices or charges are otherwise established or prohibited by statute, Executive Order, or regulation. In addition, the Department has established exceptions for assessing the FAC for small business concerns; institutions of higher education; nonprofit entities, State and Local Governments; and certain research, development, testing, evaluation, training, and exercises directly related to specified activities and approved by the Secretary of Energy.

Generally, rates charged by UT-Battelle are based on a standard rate methodology. Under this process, rates for labor, organization burden, space pool, general and administrative, etc., are set at the beginning of the year. As costs are incurred, the difference between the actual costs incurred and the standard rate being charged is accumulated in a variance account. The variance balances are routinely reviewed, balances are distributed, and standard rates adjusted, as necessary. All ORNL employees are assigned to an organization. When a cost objective is charged for an employee's time, an indirect organizational burden cost is applied, at an hourly rate, on a causal beneficial relationship in the same proportion as direct labor hours associated with each project. The application of these types of cost and the elements that make up the rates are approved by DOE via the UT-Battelle Cost Accounting Standards disclosure statement.

The accounting system, operated and maintained by UT-Battelle, meets all contractual requirements and is fully integrated with the DOE accounting system. The system conforms to DOE Orders and is periodically reviewed by DOE. The system is also subject to routine audits by the UT-Battelle Internal Audit and the DOE Office of Inspector General.

If there are any questions or additional information is required, please contact Rakisha Beatty, of the ORNL Site Office, at (865) 576-4522 or Jerry McKeehan, of the Financial Evaluation and Accountability Division, at (865) 576-0863.

Sincerely,

Judson A. Kingman
Contracting Officer

SUMMARY BUDGET ESTIMATE

LABOR		YEAR 1										YEAR 2										TOTAL			
BADGE	NAME	HOURS	SALARY	FRINGE	OB	VAT	OH	PT	COST	FTEs	CAL MOS	HOURS	SALARY	FRINGE	OB	VAT	OH	PT	COST	FTEs	CAL MOS	HOURS	COST	FTEs	CAL MOS
938444	Gregory R Watson	710.9	75,127	32,604	16,100		48,869	-	172,699	0.39	4.69	682.0	75,067	32,954	15,752		48,846	-	172,619	0.37	4.50	1,392.9	345,318	0.76	9.19
	Senior/Key Personnel Subtotal	**710.9**	**75,127**	**32,604**	**16,100**		**48,869**	**-**	**172,699**	**0.39**	**4.69**	**682.0**	**75,067**	**32,954**	**15,752**		**48,846**	**-**	**172,619**	**0.37**	**4.50**	**1,392.9**	**345,318**	**0.76**	**9.19**
	Labor Subtotal	**711**	**75,127**	**32,604**	**16,100**		**48,869**	**-**	**172,699**	**0.39**	**4.69**	**682.0**	**75,067**	**32,954**	**15,752**		**48,846**	**-**	**172,619**	**0.37**	**4.50**	**1,392.9**	**345,318**	**0.76**	**9.19**
TRAVEL																									
		BASE	BASE			VAT	OH	PT	COST	FTEs	CAL MOS	BASE	BASE		OB	VAT	OH	PT	COST	FTEs	CAL MOS	BASE	COST	FTEs	CAL MOS
	Domestic	2,000	2,000			56	285	-	2,341			2,000	2,000			56	285	-	2,341				4,682		
	Travel Subtotal		**2,000**			**56**	**285**	**-**	**2,341**				**2,000**			**56**	**285**	**-**	**2,341**				**4,682**		
	TOTAL		**77,127**	**32,604**	**16,100**	**56**	**49,154**	**-**	**175,040**	**0.39**	**4.69**		**77,067**	**32,954**	**15,752**	**56**	**49,131**	**-**	**174,960**	**0.37**	**4.50**		**350,000**	**0.76**	**9.19**
	FAC								-										-				-		
	TOTAL including FAC								**175,040**										**174,960**				**350,000**		

SBIR/STTR LEVEL OF EFFORT WORKSHEET

				Total Amount:
	Company Name	RNET Technologies, Inc.		$ 1,000,000
	Small Business	Research Institution	Third Party	Exclusions
SMALL BUSINESS LEVEL OF EFFORT				
Total Salary, Wages and Fringe Benefits (A + B)	$ 346,719			
Section D, Travel				
Section F2 - Publication Costs				
Section F6 - Facility Rental/User Fee				
Section F8, F9, F10 - Other*				
*Determine whether charges should be charged to the small business, RI, or Third Party				
Section H - Indirect Costs	$ 237,975			
Section J - Fee	$ 65,307			
RESEARCH INSTITUTION LEVEL OF EFFORT (STTR ONLY or BOTH)				
Section 5 - Subaward [Total Amount Requested by RI - (minus) Materials/Supplies, Equipment, etc.] Use RI Budget justification.		$ 350,000		
THIRD PARTY LEVEL OF EFFORT				
Section F3 - Consultant Services				
Section F4 - ADP/Computer Services				
Section F5 - Subaward				
Section F8, F9, F10 - Other*				
*Determine whether charges should be charged to the small business, RI, or Third Party				
EXCLUSIONS				
Section C - Equipment				
Equipment Costs included RI or Subaward Budget				
TOTAL				$ -
Section F1 - Materials and Supplies				
Materials and Supplies included RI or Subaward Budget				
TOTAL				$ -
Section F6 - Equipment Rental				
Equipment Rental included RI or Subaward Budget				
TOTAL				$ -
Section F8, F9, F10 - Other (Determine whether charges should be charged to the small business, RI, or Third Party)				
Other Exclusion Costs included in RI or Subaward Budget				
TOTAL				$ -
TOTAL	**$ 650,000**	**$ 350,000**	**$ -**	**$ -**

SPREADSHEET ACCURACY CHECK
Total Analytical Effort (Total - Exclusions)
1000000
Total Analytical Effort (Small Business + RI + Third Party)
1000000
Above numbers should match. If they don't something is incorrect on your spreadsheet

	Small Business	Research Institution	Third Party
LEVEL OF EFFORT	65%	35%	0%
SBIR Only Requirements (Phase II)	50%		
STTR Only Requirements (Phase II)	40%	30%	
Both SBIR & STTR Requirements	50%	30%	

Introduction

RNET Technologies Inc. (RNET) was founded in June 2003 with the goal of developing leading edge software, hardware, and electronic systems that meet the needs of the DoD, DoE, NASA, Prime Contractors and other commercial companies across the United States. We focus on research and development of advanced HPC software for Linux, Unix, Windows, and Embedded Systems. We specialize in the optimization of large scale numerical simulations, machine learning, and graph analytics codes for emerging high performance compute architectures and future exascale systems (including multi-core, many-core, and GPU based platforms) and the development of tools to improve the usability of these codes and systems. In order to address large scale computer science problems, we routinely collaborate with government laboratories, university researchers, and prime contractors. Our mission is to develop leading edge software products for our customers that enables improved usage of large scale High Performance Computing systems and large scale numerical simulations. RNET has developed numerous software and hardware components that have resulted in improvements to government owned systems, with some notable examples being optimizations to the Air Force CREATE-AV KESTREL code, the integration of GPGPU optimizations into PETSc development branch, and the development of Wattprof; a fine grained power monitoring board for emerging HPC systems. RNET, located in Dayton Ohio, was founded by Dr. V Nagarajan, an Ohio State University graduate with over 30 years of experience managing programs for various government customers (e.g., Air Force, Navy, Army, DARPA, NIST, DOE) and industrial clients.

In 2018, RNET received a DOE Phase I contract to develop a software framework that facilitates end-user verification and validation in general purpose numerical simulation packages. To accomplish this objective, RNET has developed a prototype Verification and Validation (V&V) toolkit with three key features; (1) cross library support for inserting self describing testing points into existing applications, (2) runtime test injection that allows tests to be added or removed without recompiling the code, and (3) automated documentation generation that compiles test results into a responsive, publishable html/java-scrict reports. The prototype has been tested in several of RNETs internal codes, and in the MOOSE finite element code to good effect.

RNET is pursuing a DOE SBIR Phase II contract to optimize, harden and extend the framework into a fully featured V&V toolkit that can be used reliably across a range of high performance applications. The key research objectives of the Phase II proposal will be (1) the development of efficient statistical based V&V tools that can be applied in distributed settings; (2) hardening of the injection point delivery system to ensure the code is type-safe and maintains constant correctness; (3) implementation of a workflow management system for concurrency through job based parallelism; and (4) demonstration of the toolkit in real, production quality numerical simulations. Full realization of these objectives will result in a production ready V&V toolkit that can be demonstrated and sold to customers across the numerical modeling market.

To obtain the requested SBIR Phase II grant we are submitting this commercialization plan that highlights the commercial potential of our innovation as evidenced by: (1) the significance of the technical and or economic benefits of the Phase II research, (2) the likelihood that the proposed research will lead to a marketable product and (3) the likelihood that proposed research can attract further development funding. We have given this requirement considerable thought. We have examined markets and applications, talked to potential customers and partners, and assessed the resources necessary to realize the commercial potential of the innovation. As a result of this work, we find that our proposed SBIR/STTR innovation has a high commercial potential because:

- We have a good market opportunity.
- We have a company and team that is able to realize the value of this market opportunity.
- We can establish and sustain a competitive advantage.
- We can independently finance the anticipated commercial growth.

1 Market Opportunity

The strong market opportunity is the foundation of the V&V toolkit commercialization potential. Numerical modeling and simulation (M&S) is extensively used in numerous scientific and engineering fields and disciplines such as computational fluid dynamics, high energy physics, nuclear engineering, and computational finance. Software verification and validation, the process of determining that a simulation is fit for its defined purpose, prevents erroneous simulation errors from propagating into final designs and is a fundamental, unavoidable step in the technology transfer process for numerical simulations. It is this fundamental importance that motivates our assertion of a good market opportunity for the proposed V&V toolkit.

The V&V toolkit capitalizes on a growing need for tools that facilitate the V&V of simulations designed by the end-users of highly configurable numerical simulation packages. End-user V&V is extremely important because the ultimate costs of a design failure (e.g., financial, injury, or loss of life) lies with the simulation software user; any attempt to shift the blame to the software developers of simulation library X (including the V&V toolkit) will certainly fall on deaf ears.

However, while most high quality simulation packages have robust **internal** V&V regimens, few (if any) ship with a functionality that streamlines **end-user** V&V. Instead, software simulation developers often take a legal approach, whereby the licences include language to the effect of "*use at your own risk"* or "*this software is released as-is, with no guarantee it is fit for the intended purpose"*. The V&V toolkit tool addresses the need for simple end-user validation in numerical simulation codes by providing a unified framework that facilitates the development of *explainable numerical simulations*. These explainable simulations provide the end user with not only the result, but also a detailed report on how the solution was obtained and the reasons it should be trusted. This detailed report does not shift the burden of end-user V&V to the developer, that is, and always will be the responsibility of the end-user; rather, the framework provides a mechanism for providing the end-user with a level of knowledge about the inner workings of the simulation far beyond what is provided in most numerical simulation tools.

As we will show, the V&V toolkit has a good market opportunity because:

- It addresses an important, widely prevalent problem.
- It is a good solution to the problem.
- We can immediately address several problem instances.
- We have a sound plan for addressing these instances and have made good progress on this plan.
- There are other applications of economic, social, and scientific value.

1.1 Market Problem Addressed

End-user V&V for numerical simulations solves an important problem and serves as the basis for the good market opportunity because

- ***End-User V&V is essential***: Undetected errors in numerical simulations propagate into physical designs creating issues that can cost millions of dollars to fix, cause catastrophic damage to the environment, and, in the worst case scenarios, result in the loss of human life. The Sleipner platform accident[1] is a prime example of the potential problem, an offshore platform collapsed due to failures in the finite element simulation with devastating consequences (and $700M in damage). The Sleipner accident demonstrates

[1] The Sleipner Platform Accident, by B. Jakobsen and F. Rosendahl, Structural Engineering International 4(3), August 1994, pp. 190-193.
The Failure of an Offshore Platform, by R. G. Selby, F. J. Vecchio, and M. P. Collins, Concrete International 19(8), August 1997, pp. 28-35.

how a poorly verified numerical simulations can significant consequences on business, the environment, and society. The V&V library will greatly increase the quality of end-user V&V.

- **End-User V&V is expensive to set-up:** Verification and validation of numerical simulation is a notoriously difficult task that requires a broad knowledge of the algorithms and assumptions used in the development of the simulation. Acquiring the required expertise to implement such a plan is hard enough for the developers of the simulation; let alone for customers in industry whose entire reason for using the simulation package was likely to avoid spending the millions of dollars required to develop similar functionality internally. Integration of V&V reporting tools into the software libraries will greatly reduce the burden on the end user.
- **End-User V&V requires hundreds of man-hours:** Verification and validation is an iterative process that must be re-done each time any integrated software package is updated. Software development is itself an inherently iterative process, with 3-12 monthly release schedules being common among cutting edge numerical simulation packages. As such, keeping V&V reporting up to date is in itself an expensive, time consuming task. Automation of the reporting would significantly improve the results and reduce the costs.
- **Supporting End-User V&V is difficult:** For developers, integrating support for end-user verification is also an expensive, time consuming task. The primary reason for this is that most modern software packages utilize a deep hierarchy of third party numerical simulation packages. By allowing the developer to focus on their area of expertise, these third party packages dramatically reduce the time to market. However, robust end-user V&V requires through testing at every level of the simulation hierarchy, and as such, requires the developer to fully understand and modify the third party simulations. Such requirements counteract the primary benefit of using the third party libraries, knowledge compartmentalization, and as such, make supporting end-user V&V an unenviable task.

1.2 Targeted Market Solution

The concept of an integratable V&V toolkit it is a good solution to the problem of end-user V&V in numerical simulation packages, strengthening the market opportunity.

The V&V toolkit is a C/C++ framework that promotes the development of *explainable numerical simulations.* These are numerical simulations that not only provide the user with the solution, but also a detailed report as to how the solution was obtained and why the solution can be trusted. The toolkit will include a set of highly customizable, extensible and reusable V&V tests that can be configured by the end-user at runtime to setup a robust end-user V&V regimen. All V&V testing will be self documenting, streamlining the process of re-verifying a numerical simulation each time the numerical simulation package is updated. Moreover, the V&V documentation from a simulation will be presented as a modern HTML webpage that can be viewed in any web browser and/or hosted privately (with little to no cost) on services like AWS.

The functionality imparted by the V&V toolkit will be of exceptional value to the developers of numerical simulations in the areas of finite element analysis and computational fluid dynamics. The users of the toolkit will be those developers and their customers.

The developers of a numerical simulation will use the V&V toolkit to add robust support for V&V testing at critical points in the simulation. That functionality will then be used by the simulation library to verify and validate their own simulations. The V&V toolkit could also be used as a mechanism for verifying and validating the developers own code, both as a mechanism for functional in-situ testing and as a method for generating and performing V&V for any benchmark tests that are run.

To reiterate, the V&V toolkit is a good solution to the problems described above because it provides:

- **Explainable Numerical Simulations Increase Trust in the Solution**: The V&V Toolkit provides developers with the functionality required to build explainable numerical simulations. This addresses the need for end-user verification and validation in simulation packages by allowing for the development of simulations that not only produce a result, but also a description of how the solution was obtained and why it can be trusted.

- **Runtime Configuration of Tests Reduces Setup Time**: The end-user V&V functionality facilitated by the toolkit is configurable at runtime through a single input file. This allows the users of the code to iteratively build a robust V&V regimen without ever touching the source code of the simulation.
- **A Robust Collection of V&V Tools Improves Efficiency**: The framework includes a robust collection of general-purpose V&V tools and statistical assertions optimized for performance in a large scale distributed settings. This significantly reduces the costs associated with completing end-user verification by removing the need to reimplement common algorithms for each simulation.
- **Automated Documentation Generation Reduces Time to Market**: The defining feature of the V&V toolkit is its support for automated V&V report generation. By providing the user with automatically generated, production grade V&V reports the framework reduces the burden of meeting the V&V reporting requirements.
- **Cross-library support simplifies integration**: The framework provides integrated support for multilingual, cross library V&V testing. This provides developers with a simple mechanism for facilitating end-user V&V at all levels of the simulation hierarchy. Such functionality will increase the appeal of the simulation package in the competitive numerical simulation market.

1.3 Wide Problem Prevalence

In addition to a good solution to the problem, we have a good market opportunity because the problem being addressed is widely prevalent. As mentioned above, computational simulations are a key component of the R&D pipeline in a range of industries such as computational fluid dynamics, high energy physics, nuclear engineering, and computational finance. The prevalence of smaller scale finite element analysis (FEA) software is even wider, with applications ranging from design optimization in $10 polycarbonate trinkets up to large scale parameter optimization of fighter jet wings. In all cases, the software used to inform the designs of these products must be verified and validated; and in all cases, the verification and validation of this software is a complex, time consuming task requiring input from experts across the broad spectrum of numerical simulation specialty areas (i.e., linear solvers, nonlinear solvers, finite element methods, physical domain specification, data analysis, etc.).

The primary market for the V&V toolkit is Computer Aided Engineering (CAE). Analysts at Grandview Research valued the CAE market at $6.65B in 2018[2]. Those same analysts predice the global CAE market will reach $12.8B by 2025, with a CAGR of 9.8% during the forecast period. Figure 2 shows the expected growth rates for the four main types of CAE; finite element analysis, computational fluid dynamics, optimization and simulation and multibody dynamics. The V&V toolkit supports products in all sectors of this market.

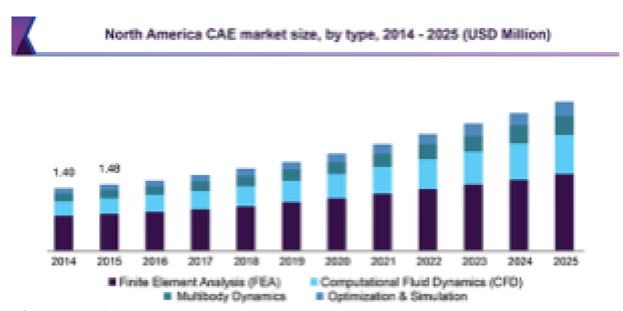

Figure 1: North America CAE market size for 2014-2025, Courtesy of www.grandviewresearch.com

1.4 Immediately Addressable Instances

In addition to wide problem prevalence, we have a good market opportunity because we can solve important problem instances upon product release (i.e., near the end of the Phase II contract):

[2] Computer Aided Engineering (CAE) Market Size, Share & Trends Analysis Report By Type (FEA, CFD, Multibody Dynamics, Optimization & Simulation), By Deployment Model, By End Use, And Segment Forecasts, 2019 - 2025,

Grandview Research, 2019, Report ID 1-68038-802-2

- **The NEAMS toolkit**: The initial targeted instance will be the NEAMS (Nuclear Energy Advanced Modeling and Simulation) toolkit that is developed by the Department of Energy's (DOE) Office of Nuclear Engineering (NE) . The NEAMS toolkit will provide "pellet-to-plant" simulation capability with an unprecedented degree of predictability for a broad class of advanced reactor and fuel cycle systems by leveraging leading edge computational methods and high performance computing technologies. The NEAMS toolkit provides a comprehensive advanced reactor simulation suite that is state-of-the-art in the nuclear reactor design community. The contentious nature of nuclear energy research led to an in depth specification for software verification, validation and design accreditation in the industry. Despite implementing a robust V&V plan, the NEAMS tools still suffers from the issues associated with end-user V&V (as evidenced via repeated calls for "automated solutions for end-user numerical simulation verification" in DOE SBIR solicitations (2018.2, 2019.2, etc) and through discussions NE program managers).
- **OpenFoam**: OpenFOAM is an open source CFD package developed primarily by OpenFoam Inc. It is independently tested by ESI-OpenCFD's Application Specialists, Development Partners and selected customers, and supported by ESI's worldwide infrastructure, values and commitment. OpenFoam represents a well used, well supported open-source package that RNET can use to demonstrate the exceptional value of the V&V toolkits explainable numerical simulation tools.

By addressing these specific instances, RNET will be able to prove to potential customers that the V&V toolkit is a robust, feature packed toolkit that can be efficiently integrated into a production grade product as a means for providing self verifying, self explaining simulation tools to their end-users.

1.5 Market Entry Plan: Direct Integration into Existing Simulations

A sound plan for addressing the initial targeted instances further improve the market opportunity. The plan includes 4 stages; with the initial market entry stage described here. The entry plan for the V&V toolkit will be to establish two to three integration contracts, whereby RNET directly integrates the toolkit into the customers existing numerical simulation tools.

These reference accounts will be used to establish sustainable positive cash flow while also allowing RNET to fully battletest the framework in real world applications. In addition, these reference accounts are important because they foster a level of trust in the toolkit and, most importantly, provide proof that the solution is not simply "vaporware" and can be used reliably in production environments.

RNET will seek out these contracts by presenting at industry specific conferences and workshops. These meetings provide a convenient vehicle to market to developers, decision makers, and stakeholders of advanced numerical simulation tools. In each case, the NEAMS support for end-user V&V will be used to demonstrate the tremendous value of the toolkit. RNET has attended these meetings in the past, and will utilize these opportunities towards the end of the Phase II project.

This is a sound entry plan because it allows RNET to establish sustainable positive cash flow while also allocating time to cement our position as the leading supplier of end-user V&V functionality and provides a mechanism for addressing the bugs and issues that are inevitably associated with early adoption.

1.6 Subsequently Addressable Market Opportunities

In addition to a sound and realistic plan for addressing initial problem instances during market entry, additional (progressively larger) market opportunities will be addressed once we have secured our initial market position. These additional addressable markets ensure that the market opportunity is sufficient to sustain the product and company. After the initial market entry (stage I), three additional stages will be addressed (commercial V&V toolkit sales, dual licensing (open source and commercial) and integration into a wide range of software libraries, and extension of the V&V toolkit into other areas such as parallel debugging and profiling). These market opportunities include:

- **Commercial Sales Toolkit and Support Services**: Once the value of the toolkit has been proven, RNET will make the V&V toolkit available to select companies and research organizations through support and training contracts. The primary benefit of this approach is that the exposure to different code bases will provide a mechanism for weeding out unforeseen bugs prior to the wide scale release of the software. The V&V toolkit has application in scientific and engineering fields and disciplines such as computational fluid dynamics, high energy physics, nuclear engineering, and computational finance. Additionally, based on feedback from customers, this stage will allow us to refine the user-experience of the toolkit. At the completion of this stage, we will have a well tested, well documented software package that has been thoroughly tested using multiple code bases and attack angles. Wide scale marketing of the toolkit through attending and presenting technical conferences, attending academic meetings and submitting journal publications will be complete during this phase of the commercialization plan.
- **Open Source and Commercial Licensing**: The core functionality of the V&V toolkit will be released under both a commercial license and an open-source source license (e.g., GPL). This will allow the software to be integrated into other Open Source projects, while requiring a license for commercial integration and distribution. The open source license will allow users to freely use the V&V toolkit in private and academic codes, helping adoption of the V&V toolkit and allowing users the ability to learn the value of the product (support contracts will also be available). Additionally, integration into Open Source tools opens up the possibility of selling end user licenses to Open Source simulation users. A one time commercial license fee will be required to integrate the V&V toolkit into commercial software, and recurring user license fees will be charged for each end-user of the third party software that includes V&V toolkit features. The one time licensing fee will include integration support. Releasing the core functionality of the software as open-source is expected to drive the uptake of RNETs support services. The specific benefits of this open-source model are that: (1) it fosters a community of developers and users that, in turn, provide free marketing for the product (2) it acts as a mechanism for allowing testing of package prior to purchasing a support contract, and (3) it allows for quick proliferation of the toolkit into a wide range of academic codes, creating a wide user-base that will lead to a demand for V&V tools in more commercial codes.
- **Expansion into Parallel Debugging and Profiling**: Issues associated with debugging and optimizing large-scale numerical simulation have long been an issue in the high performance computing community. To that end, there are a huge number of HPC debuggers and profiling libraries (e.g., TotalView, Tau, Vampir, HPCToolkit, etc.). In terms of optimization, the V&V toolkit provides a highly configurable, intuitive mechanism for inserting profiling code directly into an executable. In addition to providing an efficient mechanism for the tried and true method of debugging - the printf statement - the robust set of statistical tools provide an effective means of asserting the state of variables distributed across hundreds of thousands of cores. In fact, the statistical methods to be developed in Phase II were first developed as a mechanism for distributed relational debugging. In many respects, parallel debugging can be thought of as a less-formal, human in the loop version of V&V. Further research into runtime streaming of V&V test results, after success in the V&V market, will lead to a natural transition to, and penetration of the parallel debugging market.

1.7 Broader Social and Economics Benefits

The confidence instilled by an automatic toolkit generated V&V regimen will act as a bridge between the cutting-edge simulation technology and systems level engineers in industry, thereby increasing the rate of technology transfer for commercial, government, and academic based R&D codes. The integration of cutting edge simulation tools in the deering into the systems engineers design pipeline will lead to improved designs, faster time to market and improved safety, while also ensuring the public benefits from the nations significant investment in advanced simulation capabilities.

2 Company/Team

In addition to a good market opportunity, we can achieve the required level of commercial success because we have the company resources and team able to realize the value of the market opportunity. RNET Technologies (RNET), located in Dayton, Ohio, was founded in June 2003 as a “C” Corporation organized under the laws of the State of Delaware. RNET is a C-Corp with a personnel structure that consists of the President at the top, and project managers and engineers at lower levels. The overall mission of RNET is to develop leading-edge (embedded, system, and application) software, hardware, and electronic systems that meet the needs of the DoD, DoE, NASA, Prime Contractors and other commercial companies across the U.S., as well as the private, domestic, and international sector. Our core expertise is related to the design and development of “systems” with a focus on High Performance Computing optimizations and usability. The high-level mission of RNET is to develop commercial products that are initially based on government funded research, but can be grown into significant products used by government and private entities.

Over the years our founders have created a company and team able to realize the full value of our market opportunity of our SBIR/STTR innovation because:

- We have a good company foundation
- We have the right team
- We have already made good progress toward the V&V toolkit

2.1 Good Company Foundation

As mentioned, one reason we can realize the full value of our market opportunity, is because RNET has a good company foundation.

Kestrel and PETSc optimizations are two examples of integration of RNET optimizations into government owned codes. We have made optimizations to Kestrel, an Air Force CFD and CSD simulation code; including the CFD solver, CSD solver, and fluid structure interaction for modern HPC architectures. These were shared with the Kestrel team and integrated into Kestrel. RNET had also collaborated with the Ohio State University and Argonne National Laboratory to develop PETSc extensions targeting petascale architectures, and these extensions were included in the PETSc development branch.

Some of the hardware systems we have designed and built include the “user-programmable” 10 Gigabit Ethernet NIC (network interface card) named “SmartNIC” along with several modules of application-level and network-level offload engines for HPC applications. Part of the development of the “OCTEON Plus” NPU-based 10 GigE SmartNIC was funded by DOE SBIR/STTR projects. In addition, RNET has invested an additional $60K and built about 25 SmartNICs. We have also made an additional investment of about $70K for designing and building 10 OCTEON-II based next generation SmartNICs, to be marketed to HPC users.

We have also developed a “fine-grained” Power Monitoring system (called WattProf) consisting of a custom-designed FPGA board containing various analog to digital converters, memory and storage, along with the required firmware and data-acquisition software for HPC applications. WattProf will be useful both for large government owned HPC centers and private data centers. Initial WattProf commercialization included a $100k contract to extend WattProf to integration with Sandia National Laboratory libraries.

As part of another DOE project RNET developed the HD TomoGPR, a novel below ground imagining system for fine root analysis using tomography and Ground Penetrating Radar. RNET is currently organizing the sale of the initial prototype with PNNL. Finally, as part of another DOE project, we developed optimizations to BigData platforms (iNFORMER). Sales opportunities for iNFORMER are being explored with major Big Data players including Lexis Nexis and Hortonworks. We have recently been awarded the Yarn Ready certification for a Map Reduce like API and are pursuing market release strategies with Hortonworks.

Revenue Type	2016	2017	2018	2019
Product, Consulting, and Contract R&D Sales	$77,098	$2,175	$0	$0
SBIR Grants	$1,876,078	$1,113,170	$2,182,189	$1,500,000
Non-SBIR service contracts	$167,331	$144,786	$7,670	$11,000

2.2 Right Team

In addition to a good company foundation, we can realize the value of our market opportunity because we have the right commercialization team. Currently RNET has six full-time employees, all with graduate degrees and two with industrial experience. Of these, four full-time employees have Ph.D. degrees. The remaining employees have Masters degrees. RNET employees are some of the best in their area of expertise, and have first-hand experience with the state of the art technologies. As a team, the employees at RNET are capable of designing innovative high-tech software products that serve national interests in their matching areas of expertise.

Dr. V. Nagarajan is the President of RNET and will supervise the commercialization operation. He has a Ph.D. in Operations Research and Computer Science and an M.B.A. from the Ohio State University. He has over 30 years of research experience, managing programs for various government customers (e.g., Air Force, Navy, Army, DARPA, NIST, DOE) and industrial clients. As the official responsible for business development, he will work on the overall strategy for successful commercialization of the V&V toolkit. Currently, Dr. Nagarajan is developing strategies for collaborating with universities and national laboratories to develop and commercialize R&D ideas, and industries to develop product marketing plans. He is also interacting with potential investors to raise funding for product commercialization.

Dr. Ben O'Neill will be the PI for the project. He has a Ph.D. in Applied Mathematics (with a focus on distributed HPC computing and parallel time integration) from the University of Colorado at Boulder. Dr. O'Neill will act as the technical lead for the project. From a commercialization standpoint, his role in the project will be to facilitate communication between the end-users of the code and the development team to ensure that the V&V toolkit meets the needs of the community.

Dr. Gregory Watson (Research scientist, ORNL) has been brought in as a collaborator. Dr. Watson is the primary developer of the Eclipse Parallel tools platform, a software package that has seen wide scale uptake in applications across academia and industry. His contacts and experience gained developing and driving uptake of this platform will be extremely important.

RNET also has a working relationship with TideMark Group Limited in Columbus, Ohio. TideMark Group identifies market opportunities; locates, qualifies, and develops relationships with suppliers and/or business partners; develops strategic plans for the export, import, or capitalization of trade opportunities in identified markets; and then implements these plans. RNET has worked with TideMark in the past, and will pursue a similar relationship for the V&V toolkit if needed.

2.3 Good Progress: V&V Toolkit

Finally, we can realize the value of our market opportunity because we have already made good commercial progress. This statement is largey backed by the successful development of a functioning proof of concept during the Phase I award period. The key milestones achieved so far:

- **Jul 2018**: Obtained SBIR Phase I funding for the V&V toolkit.

- **Dec 2018:** Version 0.1 of the V&V toolkit is completed.
- **Feb 2019**: The V&V toolkit is successfully tested in simple finite element applications.
- **April 2019**: Set up collaboration agreement with ORNL for the R&D and testing in the Phase II project.

2.4 Intellectual Property/Competition

In addition to the right company resources and people, we can achieve the required level of commercial success because we can establish and sustain a competitive advantage.

While there do exist several software packages with integrated V&V procedures, there does not yet exist a generic, cross library toolkit that provides anywhere near the full array of functionality required by end users and provided by the V&V toolkit. In vendor tools (e.g., ANSYS, Altair, Fusion360, etc.), verification and validation of the overall package is par for the coarse, however, support for end-user verification techniques such as mesh refinement studies or the method of manufactured solutions is limited. When these tools do exist, they are application specific and cannot be applied across libraries. Many open-source tools have integrated V&V procedures; but again, these tools are application specific and are not implemented consistently across all levels of the applications software hierarchy. Therefore, these tools do not meet the industry wide V&V requirements.

The runtime specification of generic V&V tests, combined with automated post processing of the results is a defining and novel functionality of the V&V toolkit and has no clear open source or commercial competitors. These product capabilities enable us to establish and sustain competitive advantage because:

- We offer extraordinary value.
- We have protected our intellectual property.
- We have superior technology.
- We can manage competitive rivalry.

2.5 Extraordinary Value Proposition

We can establish a competitive advantage partly because we offer customers extraordinary value. Simply put, numerical software should not, under any circumstances, be used to inform real world design decisions without a robust simulation verification and validation plan. To that end, the proposed V&V toolkit provides customers with extraordinary value because :

- **It Reduces Time To Market**: The V&V toolkit is a huge value add from the perspective of developmental efficiency, a fact that will inevitably reduce the time to market for simulations that use the toolkit. Integration of the V&V tools into simulation code allows it to be used as part of the standard benchmark testing suite. As such, it provides a robust, self documenting approach to generating benchmark based V&V testing reports that are automatically updated each time the software changes.
- **Increases Appeal in a Competitive Market**: The web-based format of the V&V test results provides a modern, highly informative V&V report that instills a level of trust in the solution that is not available using standard approaches. This will increase the appeal of the simulation package in a very competitive market.
- **It Reduces Support and Training Requests**: The explainable numerical simulations produced by V&V equipped simulation packages provide a detailed report regarding every aspect of the simulation process; from what algorithms were used, through to a sensitivity analysis. Through this report, users will be able to immediately detect and fix any issues in the configuration. In turn, this will significantly reduce the number of support inquiries relayed back to the simulation software developer.

2.6 Intellectual Property Protection

In addition to extraordinary value, we establish competitive advantage because we have sound intellectual property protection for our innovation. During the early stages of development, RNET will use trade secrets and non-disclosure agreements to protect its intellectual property. Once our position in the market has been cemented, RNET will pursue an open-source distribution model where by the core functionality is made freely available. Releasing the core components as open-source is a large, but necessary intellectual property risk designed to capitalize on the goodwill and market penetration that can be gained through open source distribution in academic and governmental settings. Integration of the V&V toolkit into a huge range of smaller academic R&D codes is expected to drive interest and demand for RNET's V&V integration services and licenses in well funded commercial simulation projects.

This is a sound, effective and realistic plan because it balances the needs for IP protection with the market's desire for open source software. In particular, it provides us with enough time and IP protection to build a collection of reference accounts that demonstrate the exceptional value of the V&V toolkit and the support of integration services provided by RNET.

2.7 Manageable Technological Threats

We see no unmanageable threats to the superiority of our technology. This, in addition to sound intellectual property protection, will allow RNET to establish a clear competitive advantage.

As stated above, the uniqueness of the V&V toolkit lies in its uniform, cross library, and cross language API for developing self describing, explainable numerical simulations with built-in and highly configurable mechanisms for V&V. We currently see no alternative product, technology, or innovation able to achieve higher performance or effectiveness. However, the following commercial tools and academic research are being viewed as competitors, although none exactly match in application support, function, and usability.

- **<u>National Instruments (NI) TestStand</u>**: The NI TestStand software provides a modular architecture with well defined components to address the various needs of a software testing system. The modularity of the testing components aids in V&V efforts by allowing the customer to define the specific components needed in each simulation and allows each component to be analyzed independently. TestStand process models handle test functionality that is not specific to a unit under test, including UUT tracking, report generation, database logging, and batch/parallel testing. The process models that ship with TestStand are complex, so making changes to these models requires a significant validation effort. TestStand is an impressive, commercially supported testing framework and interface with a good industry reputation; however, testStand is primarily targeted toward software developers with no emphasis on numerical simulation end users, and as such does not pose a significant threat to the V&V toolkit. In particular, TestStand provides no specific focus on the needs of the numerical simulation community, and, in many ways, represents the exact problems with applying software development testing frameworks in a numerical simulation setting that the V&V toolkit is addressing.
- **<u>SimScale</u>**: SimScale delivers a trusted computer aided engineering (CAE) platform. Through verification, SimScale ensures that the problem was conducted properly. Through validation, they take it to the next step to ensure that the results reflect reality. On a regular basis, multiple validations are performed, where obtained results are compared either to analytical or experimental data. SimScale offers computational fluid dynamics (CFD) validation, finite element analysis (FEA) validation, and thermal validation. SimScale is used by numerous Fortune 500 companies, and has reached over 50,000 users worldwide with estimated revenue of $6.5 million. However, the verification and validation tools available in SimScale are application specific and can only be used inside SimScale codes. Although we are not privy to SimScale source code, it would likely be a huge, if not impossible, task to extract that V&V functionality into a generic set of tools that would rival the functionality of the V&V toolkit. In fact, SimScale represents a software package that, during early development, could have benefited from the automated V&V testing afforded by the toolkit. Moreover, we feel the explainable simulations capabilities achieved through integration of the V&V toolkit would provide tremendous value to the users of SimScale, even on top of the robust V&V plan already in place. Thus, RNET sees SimScale not only as a rival, in that they have the experience and expertise required

to develop a product similar to the V&V toolkit, but also as a potential customer. A similar argument holds for other popular CFD and FEA frameworks such as Altair Hyperworks and ANSYS Fluent.

- **Numalis**: Numalis provides tools for validating and optimizing sensitive industrial programs headed for critical sections such as aeronautics, aerospace, and defense. Numalis offers a solution for V&V called Spoat. The current range of Spoat products is composed of Spoat-Vulnerability (automatic detection of numerical issues) and Spoat-Trust (builds a reliable benchmark for numerical functions). The goal of Spoat-Vulnerability is to find vulnerabilities on variables, blocks, functions, and modules of the code. Vulnerabilities can be a drift in the numerical accuracy, unstable code, or a risky cast. To do so, Spoat-Vulnerability relies on dynamic, static, and statistical analysis (if used alongside Spoat-Trust). This represents functionality that developers would be able to replicate using the V&V toolkit when defining a new V&V testing tool. We do not see Spoat as a technological threat to our competitive advantage because it lacks the core functionality of the V&V toolkit -- explainable, self documenting numerical simulations that can be tested and analyzed at runtime. Of all the competitors, Numalis is best positioned to develop a rival product, however we feel that our staged IP protection plan and releasing the core software open source will act to mitigate this risk.
- **Open Source C++ Unit Testing Frameworks**: Literally hundreds of C++ unit testing frameworks are a potential competitor to the V&V toolkit that are available (with Google-test being the most prominent). However, these unit testing frameworks are a competitor only in the sense that they provide a means for implementing V&V of software using an approach to software verification (unit testing) that is extremely popular across enterprise and academic software development. We expect the V&V toolkit to be used in conjunction with a standard unit testing framework and expect few technical challenges from such frameworks; however, it would be remiss to omit the current status-quo as a potential barrier to commercial success.

Competitor	Markets Served	V&V Toolkit Differentiator
NI TestStand	Software developers, web, ui, enterprise	● Built with numerical simulation in mind. ● Runtime configuration for in-situ V&V testing. ● Robust I/O for creating explainable simulations. ● Automated V&V report generation with built in support for 2D and 3d visualization.
SimScale (and various other CAD/CAE Vendors)	Computational fluid dynamics, finite element analysis, structural analysis	● Automatic report generation. ● No vendor lock in. ● Supports open source tools.
Numalis	Code validation and optimization in aerospace, aeronautics, defence, etc	● Runtime test specification and configuration. ● Generates a responsive, publication grade HTML V&V report ● Enables development of explainable Numerical simulations.
C++ Unit Testing Frameworks	Software development, numerical simulation	● Specifically designed to suit the needs of the numerical simulation community. ● Tests that can be configured at run-time and run in-situ without the need to recompile ● A simple markdown syntax for presenting the V&V report.

2.8 Manageable Non-Technical Threats

In addition to sustainable technological superiority, we can establish a competitive advantage because we can manage non-technological threats from rivals.

The primary non-technological threat is the recognized expertise of and prior purchase of rivals in the fields of software development. In this regard, RNET has identified Kitware, a U.S based company that substantial R&D with researchers and academics at government labs and universities as a key rival. Kitware have achieved widespread usage of their Cmake build system, and Paraview and VTK based software visualization tools. In fact, the Phase I prototype of the V&V toolkit integrates with the Kitware open source VTK.js framework for enabling 3D scientific visualization in web based V&V reports. Kitware also played a part in the development of ADIOS, the IO system used in the V&V toolkit for writing the data to file. Kitware does not currently have any V&V tooling; however, their expertise in efficient scientific visualization positions them as a company that could potentially develop a similar product. As with Numalis, this risk will be mitigated as best as possible by using the proposed staged IP plan. The elements of the V&V toolkit that use the Kitware tools, namely an extension to the markdown syntax for processing and presenting scientific data, will be released as part of the open-source toolkit components.

Another non-technical threat is the reluctance to becoming an early adopter of the V&V toolkit. The toolkit will reach its true potential when it has been integrated into tools spanning the numerical simulation hierarchy, from low level linear algebra packages like PETSc, up through advanced multiphysics packages like MOOSE. The proposed three stage V&V market entry plan is designed to mitigate this risk by allocating time and resources during the first two years of commercialization to building a diverse set of reference accounts that fully demonstrate the value of the toolkit. Thus, we see no unmanageable non-technological rival threats.

3 Finance and Revenue Model

In addition to a good market opportunity, the right company resources and team, and the ability to create and sustain a competitive advantage; we can achieve commercial success because we have a sound and realistic finance and revenue model.

The short term goals for the product are to generate sufficient revenue to cover the robust testing and integration required to demonstrate the value of the V&V toolkit and to allow for the development of advanced V&V features. Software V&V is a requirement that is not going away anytime soon; thus, achieving these goals will allow the V&V toolkit to claim the position as the de-facto standard for an integrated V&V testing environment. The CAE and simulation market supported by the V&V toolkit is estimated at $6.65B in 2018 with a market growth rate of 9.8% year over year as described in Section 2 (giving a 2019 market estimate of $7.3B).

To fund these goals, RNET will use a combination of support contracts, government contracts/grants, product licensing sales, and internal funding. The finance and revenue model that will be used to commercialize the V&V toolkit is sound and realistic because we have a soundly staged commercialization plan, have realistic cost estimates for each stage, can cover the cost of each stage, can provide a good return on Phase II investment, and can acquire additional funds as necessary.

3.1 Soundly Staged Plan

We have a sound and realistic finance and revenue model because we have a sound, four stage commercialization plan as outlined below.

- **Reputation Generation**: The first stage will begin during the final year of the Phase II award and will include procuring non-SBIR government grants/contracts for integration of the V&V toolkit into existing government tools. As outlined in the market entry description, the goal of Stage I will be to generate sufficient contract based research funding to support robust testing and integration into real world codes. Extensive marketing efforts will also begin in Stage I. Historically, we have seen little benefit from advertising advanced technology such as this using standard paid advertising channels (magazines, internet advertisements, etc.). Rather, the toolkit will be advertised through direct communication with developers, scientific publications, and technical presentations and tutorials at conferences.

- **Integration and Support**: Stage II will focus on support and integration contracts for industrial and government users, and release the V&V toolkit as a stand-alone product for integration into simulations. The goal of this stage is to generate revenue to cover the costs of the support services offered, to develop new tools and functionality, and to keep the code up to date with the latest HPC advancements.
- **Distribution**: At this point in the project, the V&V toolkit will be a robust, well tested framework that has been demonstrated to work in a number of academic and enterprise applications. Further research and development funded through revenue from Stage I and II will help cement the toolkit as best in class. In Stage III, RNET will release the core components of the toolkit to the general public as open-source software. An open-source distribution will ensure wide scale uptake across academia that will drive demand for the V&V toolkit functionality in large commercial applications. Revenue will be generated from the sale of support and training services. The ultimate goal for this stage is to get the V&V toolkit integrated into a commercial tool such as SimScale, Fusion 360, and/or Hyperworks.
- **Expansion**: The final stage is to modify the V&V toolkit to suit validation needs in other areas like AI and server side software development. A recharacterization of the toolkit as a full software testing framework, including integration of a unit testing framework and the development of automated testing functionality will be pursued during this expansion. Funding for these expansions will be provided by new and on-going integration contracts and through support contracts. A similar, staged commercialization plan (steps I-III) will be used for bringing the V&V tools developed in these new areas to market.

This plan is soundly staged because:

- **It promotes reliability**: Releasing an incomplete and poorly tested software package too early in the design pipeline results in a poor user reputation and can open the door for other companies to develop a competing, potentially better functioning product. It would be naive to assume that we can account for every use case and issue within the initial Phase II project, thus, the initial focus on support and integration contracts is appropriate because it provides a large (3 year) self-sustaining window of revenue generation where we can complete testing and develop unforeseen enhancements required to cement the toolkit as best in class, while also mitigating risks associated with the development of rival products.
- **It builds a solid reputation**: The largest barrier to market for any simulation technology is customer uptake. The proposed plan will slowly build a reputation as the best in class for the V&V toolkit by demonstrating the value of the product and our services in a growing set of reference accounts. These reference accounts will then be used to drive further sales of support and integration contracts.
- **It suits the market**: The open-source, contracts based model is a common approach to the commercialization of advanced technologies like numerical simulation software. The approach is successful because it balances the communities desire for free and open-source software, with the need to profit from the project. When done correctly, this approach fosters a community of end-users that are often willing to provide feedback, track bugs, and commit fixes. It is that user feedback that will keep the toolkit on the cutting edge and allow us to adapt to new and emerging architectures and challenges.

3.2 Realistic Cost Estimates

In addition to having a soundly-staged commercialization plan, we have a sound and realistic finance and revenue model because we have realistic estimates of the cost for each stage. The projected expenses for the first five years of the project are outlined in the Pro-forma. RNET uses a OH rate of 54% for all employees and a G&A rate of 7%. These figures were obtained under the following assumptions (see expenses in the Pro Forma):

- One technical sales and marketing employee will be hired in 2021 at a salary of $120K in accordance with similar positions in the Dayton area. RNET will allocate 50% of this employees time to the V&V toolkit sales and marketing, and 100% thereafter. One additional sales staff will be hired in 2023 at a salary of $120K.
- One technical support person will be hired in 2020 with a 50% allocation to this project during 2020 and 100% thereafter. The base rate for a technical support member is estimated as $80K. An additional technical support allocated to the project in 2021.
- Research and development will be covered entirely through SBIR funding during the first two years. RNET will allocate 1, 1.5 and 1.875 research scientists to R&D in 2021,2022 and 2023, all at a base rate of $120K.
- RNET expects a facilities cost of approximately $10,000. Legal expenses are also needed to support our interactions with our legal counsel, partners and USPTO, including licensing and IP negotiations with our partners as well as patents and trademarks where necessary. The amount of legal services required will increase based on sales, and include initial effort for an extensive patent search and other initial legal expenses (EULA creation). RNET's G&A expenses are 7% of sales, R&D, facilities, and legal expenses.

These cost estimates are realistic cost estimates because they are largely focused toward growing and retaining a competent team capable of successfully commercializing the product, both from a technical and marketing/sales perspective.

3.3 Revenue Streams

The V&V toolkit includes five revenue streams; end user support contracts, commercial integration licenses, commercial user contracts, commercial integration contracts, and R&D contracts. All prices will be fully evaluated during Phase II and modified as appropriate.

- End user support contracts provide support (product updates, 24 hour email support, bug fixes requests, etc.) to users of the V&V toolkit, including open source developers, commercial users using the V&V toolkit on internal products (i.e., not for sale or distribution), and users of open source products that leverage the V&V toolkit (e.g., NEAMS users who desire support). The current expected annual price is $5,000 per user. The price fits inline with others in the industry, including the NI TestStand which offers its product and full support package at $4500 per year.
- Commercial integration licenses allow the V&V toolkit to be integrated into commercial code and to be distributed in commercial products (as opposed to Open Sources licenses that require any changes or integrated products to be Open Sourced). This license includes a fixed number of user licenses and integration support. The current expected one time cost (per major product release) is $75,000.
- Commercial user licenses require a fixed cost (per user) royalty. A commercial V&V user license is required for each licensee of client software that includes V&V toolkit features (e.g., ANSYS FLUENT users). It is possible that toolkit customers (e.g., ANSYS) will provide their users with V&V features only in premium products, or as a separate add on license fee. It is expected that royalty licenses fees will be negotiated, initially fees are expected to be $1,500 per user. This is a reasonable cost as simulation software licenses can range from $1,000's to $100,000+ and V&V is a critical component for many users.
- RNET will perform integration of the V&V toolkit into commercial or government codes using integration contracts. The cost of these contracts will vary based on the application complexity. An initial estimate is $250,000. The price per contract was determined under the assumption that integration will take a research scientist 8 months and is inline with contracts RNET has received in the past.
- RNET will also perform feature updates and modifications to support new simulation methodologies and to develop/support new V&V techniques via R&D contracts. It is anticipated that these contracts would be sold to government agencies or large commercial customers. An example of such a contract would be from a government agency to develop a application specific test library. The average price of $225,000 was used in accordance with other short-term government R&D contracts, e.g., SBIR Phase I awards.

3.4 Covered Costs

The first five years of projected operating earnings covering the full three stage financial plan are summarized in the Revenue Section of the Pro Forma. These estimates are based on the following assumptions of revenue generation during each stage:

- **2019:** RNET will rely entirely on SBIR funding for 2019. After expenses and taxes, the V&V toolkit is expected to have an operating margin of 7% in 2019; all of which will be reinvested into R&D in 2020.
- **2020:** The first commercial sales are expected in 2020 in the form of a single integration contract. After expenses, RNET expects a $113k profit in 2020.
- **2021:** We project that we will obtain 1 additional integration contracts in 2021. We also expect 25 third party "royalty" users (resulting from the first integration), and 10 V&V toolkit user support contracts. RNET also expects one commercial licence and one R&D contract for extending the toolkit. The operating margin for 2021 is expected to be 8.6%.
- **2022:** We will look to ramp up our integration efforts. We expect to have 200 third party (royalty) users and 50 V&V toolkit users. RNET will look to obtain two integration contracts, sell two commercial licences and acquire one R&D grant/contract. The operating margin is 22%.
- **2023:** RNET projects it will sell two integration contracts, 1 R&D contract, and two commercial licenses in 2023. Additionally, we are projecting 1500 third party users and 150 V&V toolkit users. At this point the majority of the revenue is from users licenses and services. The operating margin for 2023 will be 59%.

The number of potential users is derived using a bottom up approach starting from the nuclear engineering community (our initial market). There are 16,800 nuclear engineers in the United States[3], and ANSYS (a major simulation provider) estimates that about 10% of engineers requires simulation tools[4]. We expect that there is a larger percentage of nuclear engineers that would benefit from simulation compared to the general engineering pool. However, this will serve as a good lower bound. Therefore, it is estimated that there are at least 1,680 nuclear engineers who would benefit from the V&V toolkit. Further, about 6% of ANSYS sales are in the energy sector (of which Nuclear Engineering is a subset)[5]. Therefore, we can assume that Nuclear Engineering is less than 6% of the simulation users. Therefore, we project that there are over 28,000 potential users of the V&V toolkit. We are targeting slightly over 5% penetration (1,650 total user licenses) by 2023.

3.5 Good Return on Investment

Based on our ten year projections, the DOE investment multiple will be 9.7 and the Net Present Value will be $9.7M (see the attached DOE Investment multiple worksheet). The market size and growth rate presented in this table are extended from the Pro Forma. The market share in 2019 and 2020 are from the Pro Forma and the markets share increases to 0.07% in 2030 (note, the market share was originally calculated using a market build up method described above, as there is not a good mechanism to extract the niche market size for advanced V&V tools). A stabilized operating margin of 50% was used for the years beyond 2023.

[3] Bureau of Labor and Statistics, U.S. Department of Labor, Occupational Outlook Handbook, 2016-2017 Edition, Nuclear Engineers
[4] ANSYS Investor's Day 2014: Annual Report, Ansys Inc., March 2014.
[5] ANSYS: Initiating Coverage at Hold; $90 Price Target Steady operator with strong fundamentals; valuation keeps us on the sidelines," Evercore ISI, September 14, 2016

Pro Forma Income Statement (V&V Toolkit)

RNET Technologies, Inc.
For years 2019 to 2023

Market	Phase II (2019)	Phase II (2020)	2021	2022	2023
CAE and Simulation Market Size	$7,300,000,000	$ 8,015,400,000	$ 8,800,909,200	$9,663,398,302	$10,610,411,335
Rate Market growth (average)	9.8%	9.8%	9.8%	9.8%	9.8%
Sales (Revenue)	$ -	$ 100,000	$ 637,500	$ 1,425,000	$ 3,875,000
% market share	0.00%	0.00%	0.01%	0.01%	0.04%

Revenues					
Integration Contracts	$ -	$ 250,000	$ 250,000	$ 500,000	$ 500,000
Third Party User Royalty	$ -	$ -	$ 37,500	$ 300,000	$ 2,250,000
User Support Contracts	$ -	$ -	$ 50,000	$ 250,000	$ 750,000
Commercial Licences	$ -	$ -	$ 75,000	$ 150,000	$ 150,000
R&D Contracts	$ -	$ -	$ 225,000	$ 225,000	$ 225,000
SBIR/STTR Contract R&D	$ 500,000	$ 500,000	$ -	$ -	$ -
Total Revenue	**$500,000**	**$750,000**	**$637,500**	**$1,425,000**	**$3,875,000**

Expenses					
Licensing & Royalties	$0.00	$0.00	$0.00	$0.00	$0.00
Sales & Marketing	$0.00	$92,400.00	$184,800.00	$369,600.00	$369,600.00
Cost of services (technical support)	$0.00	$61,600.00	$123,200.00	$246,400.00	$246,400.00
Administrative (G&A) - 7%	$0.00	$7,168.00	$30,072.00	$56,476.00	$308,000.00
SBIR R&D	$465,000.00	$465,000.00			
Internal R&D	$0.00	$0.00	$184,800.00	$277,200.00	$346,500.00
Legal	$0.00	$0.00	$50,000.00	$150,000.00	$300,000.00
Facilities	$0.00	$10,000.00	$10,000.00	$10,000.00	$20,000.00
Total Expenses	**$465,000**	**$636,168**	**$582,872**	**$1,109,676**	**$1,590,500**

Operating Earnings (EBIT)					
Operating Earnings (EBIT)	$35,000	$113,832	$54,628	$315,324	$2,284,500
Operating Margin %	7.0%	15.2%	8.6%	22.1%	59.0%

Income					
Income Before Tax	$35,000	$113,832	$54,628	$315,324	$2,284,500
Tax rate	35%	35%	35%	35%	35%
Taxes	$12,250	$39,841	$19,120	$110,363	$799,575
Net Income	**$22,750**	**$73,991**	**$35,508**	**$204,961**	**$1,484,925**
Net income as %/sales	4.6%	9.9%	5.6%	14.4%	38.3%

DOE SBIR Funding: Years 0 - 2

		PH I	PHII - Year 1	PHII - Year 2
		0	1	2
($ in1000s)	Year	2018	2019	2020
SBIR Funding		$150	$500	$500
	Discount Rate		15.0%	15.0%
	Discount factor		0.93	0.80
Net Present Value (NPV)		$150	$464	$400

Market size PHII - Year 1	$7,300,000
Market growth rate	9.8%
First year of commercial sales:	2021

Ten Year Revenue Projection: Years 3 - 12

		3	4	5	6	7	8	9	10	11	12
($ in1000s)	Year	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030
	Market Size	$ 8,800,909	$9,663,398	$10,610,411	$11,650,232	$12,791,954	$14,045,566	$15,422,031	$16,933,390	$18,592,863	$20,414,963
	Market growth r	9.8%	9.8%	9.8%	9.8%	9.8%	9.8%	9.8%	9.8%	9.8%	9.8%
	Market Share	0.01%	0.01%	0.04%	0.04%	0.05%	0.05%	0.06%	0.06%	0.07%	0.07%
Gross Revenues mfg/licensee		$638	$1,425	$3,875	$4,660	$5,756	$7,023	$8,482	$10,160	$12,085	$14,290
	Royalty rate	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Gross Revenues Licensing		$0	$0	$0	$0	$0	$0	$0	$0	$0	$0
	Operating Margi	8.60%	22.10%	59.00%	59.00%	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%
Operating Profits		$55	$315	$2,286	$2,749	$2,878	$3,511	$4,241	$5,080	$6,043	$7,145
	Discount Rate	15.0%	15.0%	15.0%	15.0%	15.0%	15.0%	15.0%	15.0%	15.0%	15.0%
	Discount factor	0.69	0.59	0.51	0.44	0.38	0.33	0.28	0.24	0.21	0.18
Net Present Value (NPV)		$38	$187	$1,169	$1,211	$1,092	$1,148	$1,194	$1,233	$1,263	$1,287

Project NPV and Investment Multiple

Cumulative NPV 10yr Profits:		$9,822
Cumulative NPV SBIR funding		$1,014
Project NPV	$	8,808
DOE Investment Multiple		9.7

RNET Technologies (i.e., RNET), located in Dayton, Ohio, was founded in June 2003 as a "C" Corporation organized under the laws of the State of Delaware. The overall mission of RNET is to develop leading-edge software products that will meet the needs of the DOE, DOD, NASA, Prime Contractors and other Commercial Companies. To meet this mission, RNET is pursuing R&D and Product Development in HPC (high performance computing).

RNET received its first set of two Phase I SBIR contracts in 2004. As of November 2018, RNET has received 18 Phase II SBIR/STTR contracts of which three are on-going while the remaining have been completed. Five of the Phase II SBIR projects have been funded by the Air Force, three have been funded by NASA, one by DARPA, and the remaining nine have been funded by the DOE.

As of October 2018, RNET has received more than 40 Phase I projects from the Air Force, Navy, MDA, NASA, NSF, DARPA, SOCOM, and DOE.

In addition to the SBIR/STTR projects, RNET has also supported a non-SBIR BAA-type support contract (F8650-10-D-1750) named CIRE (Center for Innovative Radar Engineering), which focuses on "next generation" GHz and THz frequency radars. Moreover, RNET has received a Phase III contract ($100K) from Sandia National Laboratories, on WattProf HPC power monitoring product.

As a company, we have strong expertise to develop advanced software for High Performance Computing applications in several domains.

The first Phase II SBIR completed by RNET was funded by the Air Force (FA8650-05-C-4303). It developed an innovative "virtual objects based compression (VOBC)" video compression software package. We received Phase II Enhancement funding of a $100K that included $50K funding from DARPA to enhance the capabilities of the VOBC software and to investigate its use in UAVs. We were able to generate $29K revenue by conducting video processing work for a small company that was developing an innovating 3-D video technology. In addition, we also generated $337K from an Air Force funded video/image processing system for a mobile IED detection system that also used Android platform. This Android work led to generation of $4K from a research group at the Ohio State University.

The second Phase II SBIR we completed, which was also funded by the Air Force (FA8650-06-C-1019), developed 2 GHz/pixel sampling ROIC (read-out integrated circuit). Based on this expertise, we were able to work with a small company named Traycer, which was developing a ROIC/Sensor as part of a Tera-Hertz camera system. We designed and built off-chip ROIC hardware including a high-speed, low-noise signal amplifier and digitizer board, and this project generated $20K in revenues.

The third Phase II SBIR we completed was funded by the DOE (Grant DE-FG02-05-ER84163). In this project, we designed and built a 10 Giga-bit Ethernet SmartNIC. This NIC is based on the OCTEON-Plus 12 core processor from Cavium Networks. It also has 2 giga-bytes of on-board memory. The innovation is offloading and we needed to develop a range of network-level and application-level offload engines (and the required drivers and firmware) including encryption and SSL offload engines for Globus/GridFTP, and these were accomplished in the next two STTR/SBIR projects (Grants DE-FG02-08-ER86360 and DE-SC-0002182) that were completed. We have also invested about $60K in building about 25 SmartNICs. Our plan is to loan these SmartNICs to potential customers in the DOE, universities, and the Industry. We have also invested an additional $70K to develop a series of 10, 20 and 40 giga-bit SmartNICs based on the next-generation OCTEON-II processor which has 32 cores running at almost 1.5 GHz. All the offload engines that we have developed for the OCTEON-Plus processor based 10 GigE SmartNIC will also be easily portable to the OCTEON-II based SmartNICs.

In another completed DOE Phase II SBIR project (DE-SC-0002434) we have optimized the PETSc (portable extensible toolkit for scientific computing) library for emerging architectures including multi-core processors and GPUs. As part of that project, the team investigated algorithms, data structures, and techniques to enable applications based on the PETSc library. PETSc is an open source product, and thus commercial sales of the optimized PETSc library are not feasible. RNET is pursuing non-SBIR contracts from the Air Force as part of commercialization.

RNET has also developed hardware and software tools as part of DOE STTR Grant DE-SC-0004510 for fine-grain monitoring of power consumption when applications run on compute nodes in a HPC cluster. The first release of this product is ready and several evaluation toolkits have been distributed to potential customers.

On an Air Force funded Phase II STTR (FA9550-12-C-0028)) we have made optimizations to the Kestrel, an Air Force CFD and CSD simulation code. These optimizations have been included in the production Kestrel release. In addition, RNET is pursuing additional opportunities for Phase III funding to develop additional optimizations.

We are still in the early commercial stages of sveral other Phase II projects. As part of a NASA Phase II (NNX14CG06C) we are developing FPGA optimization tools using empirical tuning. We are pursuing two commercialization routes. The first is licensing through a major FPGA vendor such as Xilinx, and the second is Phase III NASA funding to apply to tool to NASA FPGA designs. An improved GUI was recently developed for this project, and released in August of 2018. As part of a DOE Phase II project (DE-SC0011322) we have developed HD TomoGPR, a novel below ground imagining system for fine root analysis using tomography and Ground Penetrating Radar. This system will be sold to domain scientists and marketed to other market segments (e.g., concrete and bridge health analysis). We recently completed an initial experiment in cooperation with DuPont/Pioneer and are pursuing a non-SBIR project to adapt the product for runway analysis. The initial prototype is being delivered to PNNL. As part of another DOE Phase II project (DE-SC0011312) we have developed optimizations to BigData platforms. Joint sales opportunities are being explored with major Big Data players including Lexis Nexis and Hortonworks. We have been award the Yarn Ready certification for a Map Reduce like API. These tools will be released on RNET's website. Finally, we are just beginning a Phase II project developing an automated linear solver selector to improve the runtime, accuracy, stability, and power of scientific simulations. This tool will be integrated into government applications and commercialize as a tool to be integrated into third-party commercial applications at companies such as ANSYS. Finally, we have recently commenced a DOE Phase II to develop CloudBench:NE, a web based simulation, provenance, and sharing workbench for nuclear engineering using NEAMS tools.

Small Business Name: **RNET Technologies, Inc.**

Grant/ Contract Number	Year of Award	Project Title	Status	Award Amount	Sales and Revenue: SP	SF	SO	LR	Total Award Sales	Investment: IS	IF	AI	VC	OI	Total Award Investment
FA8650-05-C-4303	2005	Data Compression	Completed	$ 750,000	$ 33,155	$ 437,700	$ -	$ -	**$ 470,855**	$ -	$ -	$ -	$ -	$ -	**$ -**
FA8650-06-C-1019	2006	Parallel Array Sample Hold Read Out Integrated Circuit	Completed	$ 750,000	$ 19,656	$ -	$ -	$ -	**$ 19,656**						
DE-FG02-05-ER84163	2006	Interoperability between Ultra High Speed Networks and Traditional IP Networks	Completed	$ 750,000	$ -	$ -	$ -	$ -	**$ -**						
DE-FG02-08-ER86360	2009	Creating Scalable Petascale File Systems Using Application-Aware Network Offloading	Completed	$ 750,000	$ -	$ -	$ -	$ -	**$ -**						
DE-SC-0002434	2010	Accelerating Parallel Numerical Libraries to Petascale and Beyond	Completed	$ 1,000,000	$ -	$ -	$ -	$ -	**$ -**						
DE-SC-0002182	2010	Enhancement of GridFTP Performance through Network Reservation Integration & hardware Offloading	Completed	$ 1,000,000	$ -	$ -	$ -	$ -	**$ -**						
NNX-10-C-B47C	2010	Radiation Mitigation Methods for Reprogrammable FPGA	Completed	$ 600,000	$ -	$ -	$ -	$ -	**$ -**						
DE-SC0004510	2011	HPC Application Energy Profiling for Energy Optimization	Completed	$ 750,000	$ -	$ 100,075	$ -	$ -	**$ 100,075**						
FA8650-11-C-1160	2011	Advanced ROIC Technology for SLS Photodetectors	Completed	$ 742,474	$ -	$ -	$ -	$ -	**$ -**						
FA8650-12-C-5118	2012	Fault Tolerant ROIC	Completed	$ 749,999	$ -	$ -	$ -	$ -	**$ -**						
NNX-12-C-A84C	2012	Rad-Hard and ULP FPGA with "Full" Functionality	Completed	$ 749,999	$ -	$ -	$ -	$ -	**$ -**						
FA9550-13-C-0017	2013	Scalable Multi-Tiered CFD and CSD Codes for Kestrel	Completed	$ 750,000	$ -	$ -	$ -	$ -	**$ -**						
NNX-14-C-G60C	2014	OrFPGA: An Empirical Performance Tuning Tool for FPGA Designs	Completed	$ 749,999	$ -	$ -	$ -	$ -	**$ -**						
SC 0011322	2015	Ground Penetrating Radar (GPR) System and Algorithms for Fine Root Analysis	Completed	$ 1,499,999	$ -	$ -	$ -	$ -	**$ -**						
SC 0011312	2015	iNFORMER: A MapReduce-like Data-Intensive Processing Framework for Native Data Storage and Formats	Completed	$ 1,000,000	$ -	$ -	$ -	$ -	**$ -**						
D16PC00183	2016	Performance Portable Framework for Developing Graph Applications	Ongoing	$ 1,000,000	$ -	$ -	$ -	$ -	**$ -**						
SC 0013869	2016	Automated Solver Selection for Nuclear Engineering Simula	Ongoing	$ 1,000,000	$ -	$ -	$ -	$ -	**$ -**						
SC 0015748	2017	Cloud-based Scientific Workbench for Nuclear Reactor Simu	Ongoing	$ 1,000,000	$ -	$ -	$ -	$ -	**$ -**						

SBIR.gov SBC Registration

SBC Control ID:	SBC_000000706		
Company Name:	RNET TECHNOLOGIES, INC.		
Address:	240 W ELMWOOD DR STE 2010		
City:	DAYTON		
State:	OH	**Zip:**	45459-4248
EIN (TIN):	200098466	**DUNS:**	141943030
Company URL:	http://www.rnet-tech.com		
Number of Employees:	6		
Is this SBC majority-owned by multiple venture capital operating companies, hedge funds, or private equity firms?	No		
What percentage (%) of the SBC is majority-owned by multiple venture capital operating companies, hedge funds, or private equity firms?	0.00%		

It is anticipated that all generated digital data will be protected as SBIR/STTR data and therefore will not be publicly shared during the applicable SBIR/STTR data protection period. If any data generated under this award are published, an effort will be made to also release any related digital data that is not protected SBIR/STTR data.

OAK RIDGE
National Laboratory

PO Box 2008
Oak Ridge, TN 37831-6163
(865) 574-6224| nicholsja@ornl.gov

April 18, 2019

Dr. V. Nagarajan, President
RNET Technologies, Inc.
240 West Elmwood Drive
Dayton, Ohio 45459-4248

Dr. Nagarajan:

Letter of Support for ORNL Personnel to Participate in Project: VnV: A Self Documenting Testing Framework for In-situ Verification and Validation in High Performance Computing Applications

This is to confirm Oak Ridge National Laboratory research scientist, Dr. Gregory Watson, will participate in the proposed project, "VnV: A Self Documenting Testing Framework for In-situ Verification and Validation in High Performance Computing Applications," in response to DOE SBIR-STTR FY 2019 Phase II Release 2 Funding Opportunity Announcement DE-FOA 0001976.

Pending award and all required DOE approvals for this project, Dr. Watson will provide expertise and software development support for large scale distributed tests, a type safe injection point system using LLVM, and a lightweight distributed test execution framework. Dr. Watson will also provide expertise and software development support for integrating job parallelism features of the Eclipse Parallel Tools Platform.

Dr. Watson will commit 1,393 hours during the period of performance; budget table provided on page 2. If you need further information, please do not hesitate to contact me at 865-574-6224 or nicholsja@ornl.gov.

Sincerely,

Jeffery A. Nichols
Associate Laboratory Director
Computing and Computational Sciences

JAN:ltm

c: File–NoRC

VnV: A Self Documenting Testing Framework for In-situ Verification and Validation in High Performance Computing Applications," in response to DOE SBIR-STTR FY 2019 Phase II Release 2 Funding Opportunity Announcement DE-FOA 0001976.

Estimated Budget by Fiscal Year				
Category	**FY 2019**	**FY 2020**	**FY 2021**	**Total**
Labor	$14,099	$123,913	$109,592	$247,604
Travel	$243	$2,057	$1,812	$4,112
Overhead	$5,598	$49,186	$43,501	$98,285
Subtotal	$19,940	$175,156	$154,904	$350,000
Grand Total	**$19,940**	**$175,156**	**$154,904**	**$350,000**
LDRD	$798	$7,006	$6,196	$14,000

Department of Energy

ORNL Site Office
P.O. Box 2008
Oak Ridge, Tennessee 37831-6269

April 17, 2019

Mr. Carl Hebron
Program Coordinator
Small Business Innovation Research/
Small Business Technology Tranfer
Program Office
U.S. Department of Energy
1000 Independence Avenue, Southwest
Washington, DC 20585-1290

Dear Mr. Hebron:

DEPARTMENT OF ENERGY (DOE) FISCAL YEAR (FY) 2019 SMALL BUSINESS INNOVATION RESEARCH (SBIR)/SMALL BUSINESS TECHNOLOGY TRANSFER (STTR) PHASE II, RELEASE 2, FUNDING OPPORTUNITY ANNOUNCEMENT DE-FOA-0001976 (INTERNAL REFERENCE NUMBER NFE-19-07729)

UT-Battelle, LLC is the managing and operating contractor for the Oak Ridge National Laboratory (ORNL) under DOE Prime Contract Number DE-AC05-00OR22725. As a DOE Office of Science, Federally Funded Research and Development Center, ORNL is requesting authorization to team with RNET Technologies, Inc. in response to the FY 2019 Phase II, Release 2, of the DOE SBIR/STTR Solicitation DE-FOA-0001976. ORNL has submitted to the ORNL Site Office a statement of work (SOW) to assist in the development of a generic testing framework for designing and injecting self-documenting verification and validation software components into advanced numerical simulations. This SOW has been reviewed and approved according to our normal procedures and will be performed in accordance with contract number DE-AC05-00OR22725.

Authorization is granted for ORNL to participate in the proposed effort. The work proposed for the laboratory is consistent with, or complementary to, the missions of the laboratory, will not adversely impact execution of the DOE assigned programs at the laboratory, and will not place the laboratory in direct competition with the domestic private sector.

If there are any questions or additional information is required, please contact me at (865) 576-9262 or David Myers at (865) 576-5629.

Sincerely,

KENNETH KIMBROUGH

Digitally signed by KENNETH KIMBROUGH
Date: 2019.04.17 10:57:34 -04'00'

Kenneth L. Kimbrough
Contracting Officer

DOE FY18.2 SBIR Topic 30d (Modeling and Simulation)

Phase I Final Report: DRAFT

An Extensible Verification and Validation Library with NEAMS Workbench Integration

Contract # DE-SC0018728

Ben O'Neill, PI Gerald Sabin,

[1] RNET Technologies

240 W Elmwood Dr

Dayton, OH 45459-4296

boneill,gsabin@rnet-tech.com

Contents

1 Identification and Significance of the Problem or Opportunity, and Technical Approach

This is the final report for contract DE-SC0018728, a Phase I DOE SBIR entitled "An Extensible Verification and Validation Library with NEAMS Workbench Integration". The work was performed by RNET Technologies Inc.

1.1 Significance and Identification

Numerical simulations are an essential component of the R&D pipeline, with new and more powerful algorithms and packages being developed every year. For numerical simulation codes being used to design real world nuclear reactors, erroneous simulations can result in design errors that can be extremely expensive to fix, damage the environment, and ultimately result in loss of human life. As such, the verification and validation of all numerically obtained solutions is essential, especially when those solutions are going to be used in the design of real-world products. One only needs to look to the catastrophic Sleipner platform accident, where an offshore platform collapsed due to failures in the finite element simulation, to get an idea of the consequences associated with using unverified numerical simulations.

V&V is usually seen as a one-off event that occurs once the computational model has been finished. This results in developers delaying V&V, which can increase the effort and cost required to fix errors. Additionally, it is rarely the case that a computational model is not under continuous, or at least incremental, development. In that case, each time the computational model changes (be it a change in the user driven simulation or the underlying computational toolkit) all previous verification and validation of the model becomes void. Without the updated V&V the improved simulation code is often not adopted by the end users. As such, a streamlined approach to V&V is an essential component of any design pipeline that uses numerical simulations to influence real world designs.

One of the goals of the NEAMS program is to equip end-users with a robust set of high-fidelity multiphysics capabilities that can be used to inform lower-order models for the design, analysis and licensing of advanced nuclear systems and experiments. Given the high stakes nature of nuclear power generation, it is essential that all NEAMS code, including the core tools and end-user driven and written simulations, are verified and validated using industry best practices. The NEAMS group uses internal tools and processes to verify and validate its core tools (see the NEAMS Software Verification and Validation Plan Requirement (Version 0) specification [**?**]), but, as of yet, there is limited support for end-user driven verification and validation of simulations. The proposed V&V toolkit will be designed to full this gap, providing end-users with an automated framework for verifying and validating numerical simulations based on industry best practices.

The goal of the Phase I/II project is to develop a framework that facilitates the development of *explainable* numerical simulations. Here, the term *explainable* is borrowed from the field of artificial intelligence, where researchers are looking to address issues associated with trust in AI algorithms. In the context of V&V, we define an *explainable* numerical simulation to be a simulation that, in addition to the final solution, provides a detailed report as to how the solution was calculated and why it can be trusted. To do this, the proposed framework will provide all the functionality required to create such a simulation, including support for:

- Writing a detailed V&V plan.
- Performing Mathematical and algorithmic testing (convergence analysis, mesh refinement studies, method of manufactured solutions, etc.).

- Verifying and Validating a broad benchmark testing suite.
- Performing Uncertainty quantification and sensitivity analysis.
- Comparing of simulation results with experimental data and results from third party simulations.
- Automatic documentation of the V&V effort.

The Phase I project focused on prototyping an automated V&V framework for numerical simulations that are written and driven by end-users. In particular, the Phase I effort was directed toward developing effective techniques for injecting the required functionality into general purpose numerical simulation packages. To that end, this report will introduce the reader to *VnV*: a self describing testing framework that facilitates in-situ V&V in advanced numerical simulations with the following functionality:

- Cross library and multi-lingual support for run-time test injection in numerical simulations.
- Efficient data ouput and analysis in HPC settings.
- Support for modular, generic testing libraries that can be configured at runtime.
- A simple XML configuration file that allows users to fully control the V&V process without recompiling.
- An Automated system for generating V&V reports with support for advanced data visualization techniques.

2 Phase I Work and Feasibility Study

The result of the Phase I project is a prototype of the VnV framework. As we will show, this framework provides all the core functionality required to facilitate in-situ V&V and automated documentation generation in numerical simulations.

The requirements set by the project team for the Phase I prototype were:

- Cross library support. Most modern numerical simulations rely on a deep hierarchy of numerical simulation tools. For example, BISON, the fuel performance code in the NEAMS toolkit, is built using MOOSE; MOOSE uses libMesh; libMesh uses PETSc; PETSc uses hypre; and hypre uses BLAS. Thus, it is a requirement of the prototype that the system should provide a single interface for in-situ testing in any library linked to the final executable.
- Multilingual support: The big three computing languages in high performance computing are C, C++ and FORTRAN. To ensure the system has wide applicability, it is essential that users be able to use the framework in software written in any one of these languages.
- Run time configuration: Users should be able run V&V tests without needing to recompile either the executable or any of the linked libraries. The tests themselves should be completely independent from the source code and configurable at runtime.
- Simple integration: The injection point system should be simple to integrate into existing applications.
- The framework should automatically generate a highly customizable, publication grade V&V report for each simulation.

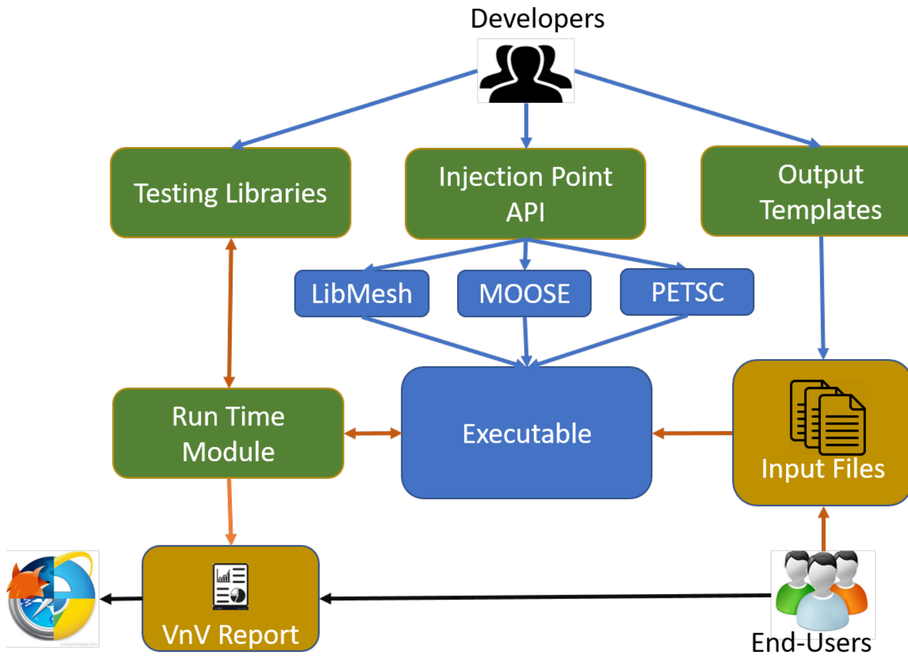

Figure 1: The VnV toolkit development lifecycle. Here, green boxes represent core VnV functionalities. Developer interactions are shown in blue, runtime interactions are shown in orange and post-processing interactions are shown in black.

These requirements were chosen because they represent the core functionality required to facilitate the development of explainable numerical simulations. That is to say, in demonstrating this functionality, the project team will be able to prove the feasibility of the proposed approach in equipping general purpose numerical simulation toolkits with built in support for end-user solution verification and validation.

Figure 1 provides an example of how developers and end-users will interact with the VnV toolkit. In this case, we show an example of how the toolkit functionality might be implemented in the MOOSE toolchain. The first step in the VnV development life-cycle is to specify and describe the injection points. These injection points will be placed at key locations of the code where testing can and should take place. In the Phase II product, inserting an injection point into a code will be as simple as annotating some variables for inspection and calling an injection point function. In addition to adding the code, developers will also complete an output template that will be used in the final V&V reports to describe the state of the simulation each time that injection point is met. The VnV tests are developed in external libraries and hence, can be developed either by the developer of the simulation or by the end-user of the code. The core framework will also include a robust set of general purpose V&V tests. We envision that the developers of a numerical simulation package will ship the library with hard-coded injection points, a set of custom V&V tests and a number of VnV configuration files.

2.1 Task 1: Develop Routines for Data Collection with ADIOS.

Work on task one began with a through investigation into the optimal approach for allowing injection points to be inserted into existing code bases. The goal of the task was to provide a simple mechanism for facilitating data collection and testing that required minimal changes to the build system, had small overheads when testing was turned off, and that which could be removed from the final executable during compilation if need be. The project team investigated several approaches to doing this including the development of a custom C pre-processor and an investigation into binary instrumentation with the Dyninst

API.

In the end, it was decided that the best approach for performing V&V data collection and testing inside existing simulations was to produce a framework that allowed the developer to hard-code injection points into the original source code. This approach is maintainable (the injection point specifications exist in the code-base), portable (it uses only standard C functions) and efficient (pointers are used to avoid copying the data). In what follows, we describe the framework developed to support data collection and V&V testing in existing simulations.

2.1.1 Injection Points

At the core of the framework are injection points. Injection points represent locations in a code where V&V testing can take place. Inserting an injection point into an existing function is a simple, three step process; (1) include the "vv-runtime.h" header file, (2) place injection points in the code and (3) write the injection point specification.

Declaring an injection point is as simple as calling the INJECTION_POINT macro. The format for this macro is:

```
INJECTION_POINT(<name>, <stage>, <type> <variable>,...)
```

The macro is expanded to a variadic C function during preprocessing. Here, the unique name represents the id that will be used to define the injection point in the configuration files and the final reports. This name must be unique across all injection points in an executable.

The stage parameter is an integer value that defines the step that this injection point belongs to. Using this parameter, the developer can set up multi-staged injection point testing. Tests defined on staged injection points stay in scope across all stages allowing tests to collect data across multiple code locations. The Phase I prototype supports up to 8000 stages at each injection point, although we have yet to come across a reason for an injection point with more than three or four stages. There are a few restrictions on the stage parameters to ensure efficient data output. First, a single stage injection point should be specified with a stage parameter of -1. A stage parameter less than 1000 should only be used to represent the starting point of a staged injection point. Likewise, a stage parameter greater than 9000 should be used to indicate that this is the last stage in the staged injection point. This allows a staged injection point to have multiple entry and exit points, while also allowing us to automatically handle recursion of injection points.

The macro supports up to 25 separate variables; although we have not come across cases where more than 5 variables are required. Specifying a variable is as simple as writing its type and name. During preprocessing, the macro expands each variable-type declaration as

```
..., <type> <name> , ... --> ..., "<type>", (void*) &name , ...
```

The runtime module of the framework then maps that void* pointer and the string based type specification to the user specified tests for processing. This is an obviously risky approach that, when implemented incorrectly, could lead to memory corruption errors. For example, it would be easy for a developer to change the type of the variable in the code, but forget to update the string in the injection point declaration. A key goal of the Phase II work will be to develop a custom pragma directive that automatically detects the correct type for each variable.

Figure 2 shows a class for evaluating a linear function that has been enhanced with a single stage VnV injection point (the stage parameter is -1).

In C++ codes, the developer can also opt to pre-register the injection point with the runtime module using the REGISTER_IP macro. This macro uses a feature of object orientated programming languages

```
class LinearFunction {
public:

    double slope, intersection;

    LinearFunction(double slope_, double intersection_) :
      slope(slope_), intersection(intersection_) {}

    double eval(double x) {
       double value = slope*x + intersection;
       INJECTION_POINT("IP_1",-1,
                      double slope,
                      double intersection,
                      double x,
                      double value);
       return value;
     }
};
REGISTER_IP("IP_1",-1,double slope,double intersection,
             double x, double value);
```

Figure 2: A code Snippet showing a member function enhanced with a single stage injection point called "IP_1". This injection point is declared inside a member function designed to evaluate a linear function at a particular value of x. This injection point provides access to the member variables describing the slope and intersection point of the linear function, the input parameter x that defines where the function should be evaluated and to the result that will be returned. A user could add a simple test to verify the accuracy of the value being returned and/or to assert that the slope and intersection are correct in accordance with the physical model they are attempting to simulate.

that allows code in the constructors of static variables to be executed prior to the main function. In this way, one can register certain code elements by simply defining a static variable. Self registration is particularly nice because it allows for run-time detection of the injection points present in the call-graph. That information allows for the automatic generation of a customized test configuration file for each executable. All calls to the REGISTER_IP macro should be made outside of function calls and classes, as shown in the last line of the code snippet in Figure 2. Unfortunately, static variables can only be initialized with constant expressions known at compile time in C and FORTRAN, so instead, information about injection points must be determined either by running the full simulation, or by pulling information from the injection point specification files. The custom compiler extension developed in Phase II will address this issue by hardwiring information about the included injection points into the meta-data of the executable or library.

2.1.2 The Testing Interface

The second facet of the framework is the V&V testing interface. The development of the test interface was based on the idea that tests should be loaded at runtime and defined independently of the source code. To achieve this, the test interface was built using an C++ plugin pattern. This pattern allows users to develop tests in separate testing modules that can be loaded and configured at runtime using an XML configuration file. This allows the users to add or remove tests from injection points located in any linked library without ever needing to recompile the executable or any of the libraries.

The first step in the development of a new VnV test is to create a testing library. The framework includes a library generation script that will automatically build the directory structure and makefiles required to build this library. Once the library has been initialized, the user can begin to develop individual tests.

Figure 3 shows the declaration for the IVVTest interface that needs to be implemented when developing a new test. To simplify the process of writing tests, the VnV framework includes a test generation script. If the library generation script was used, this script will be available in the src directory of the new testing library. To run this script, the user should provide a unique name for the test and a list of the names and types of parameters that will be supported by the new test. Using this information, the script generates all the boiler plate code required to implement the IVVTest interface and to handle all the required type-casting.

Implementing the interface is a three step process. First, the developer should implement the delcareParameters function. In this function, the developer will list the parameter names and types that will be supported by the test. For example, the test shown in Figure 4 supports the declaration of two parameters, a double named slope and a double named intercept. The user will map the injection point parameters to the test parameters at runtime using the test configuration file.

The main function of the test is the "runTests" function. The test generation script does all the dirty work required to cast the injection point parameters to their correct object types. As shown in Figure 4, the user is then required to implement a single function of the form

```
TestStatus runTests(adios2::Engine &engine, int testStage, ...);
```

Here, testStage is an integer representing the stage that the test is currently in. A test can support as many stages as there are integers. As will be shown below, the test stages are mapped to injection point stages at runtime using the test configuration file. The ... represents pointers to the variables requested by the test in the declareParameters function.

Data output can occur at any point in the runTests functions. The core runtime module ensures that each test stage is completed in a unique ADIOS2 step. This allows for efficient compartmentalization of the test output and makes post-processing significantly easier. Data output is completed directly in

```
            class IVVTest {
public:

  // Pure virtual function that must be implemented by Base Class. This function
  // defines what happens during the test. All data output should also occur
  // inside this function.
  virtual TestStatus runTest( adios2::Engine &engine, int stage , NTV &params) = 0;
  virtual void declareParameters(std::map<std::string,std::string>&parameters) = 0;

protected:

    VVTestConfig m_config;
    NT m_parameters;

    IVVTest(VVTestConfig &config);
    virtual ~IVVTest();
    TestStatus  _runTest( adios2::Engine &engine, int stageVal, NTV &params);

    template <typename T>
    T* carefull_cast(int stage, std::string parameterName, NTV &parameters );
};
```

Figure 3: The IVVTest Interface. The framework includes a test generation script that generates all the boiler plate code required to implement this interface. All the developer needs to do is implement the ”declareParameters“ function and the ”runTests“ function.

```
#include "injection.h"
#include <math.h>

class LinearTest : public IVVTest {

    bool valid = false;

    void declareParameters(std::map<std::string,std::string> &parameters) {
        parameters.insert(std::make_pair("slope", "double"));
        parameters.insert(std::make_pair("slope", "intersect"));
    }

    // The actual testing code. In this case we check if the slope and intersection
    // point are valid (>0) and write the values to file.
    TestStatus runTest(adios2::Engine &engine, int stage, double* slope, double* intersect) {

        valid = ( slope > 0 && intersect > 0 ) ? 0 : ( slope <= 0 ) ? -1 : 1;
        engine.Put("slope", *slope);
        engine.Put("intersect",*intersect);
        engine.Put("valid", valid);
    }

    void declareIO(adios2::IO &io) {
        io.DeclareVariable<double>("slope");
        io.DeclareVariable<double>("intersect");
        io.DeclareVariable<int>("valid");
    }

    // BoilerPlate Code Automatically generated by the test generation script.

    TestStatus runTest(adios2::Engine &engine, int stage, NTV& parameters ) {
        double* d0 = carefull_cast<double>(stage,"slope", parameters);
        double* d1 = carefull_cast<double>(stage,"intersect", parameters);
        int testStage = m_config.getStage(stage).testStageId;
        return runTest(engine, testStage,d0,d1);
    }
}

//More boiler plate code generated by the test generation script. This code registers
// the test with the runtime module when the shared library is loaded.

extern "C" {
      IVVTest* LinearTest_maker(VVTestConfig &config) {
        return new LinearTest(config);
    }
  void LinearTest_DeclareIO(adios2::IO& io) {
     LinearTest::DeclareIO(io);
  }

};

class LinearTest_proxy {
public:
    LinearTest_proxy(){
        // Register the test with the factory
        VV::test_factory["LinearTest"] = std::make_pair(LinearTest_maker,LinearTest_DeclareIO);
    }
};

LinearTest_proxy lt_p;
```

Figure 4: An example of a custom VnV test. In this case we implement a test that checks that the slope and intersection point of the linear function are positive and writes the result to file.

ADIOS through the ADIOS2 read/write API and the adios2::Engine. For example, the test shown in Figure 4 writes the slope and intersect points received from the injection point to file along with an integer representing the validity of those values.

The next step is to pre declare the IO variables that the test will utilize. This is optional, however, it is considered best practice because it allows for optimizations in the handling of the meta-data inside ADIOS. The final step in writing the test is to register the test. Registration provides the core runtime module with the information necessary to initialize the test at runtime. The test generation script automatically generates the code required to do this, hence, no further input is required from the developer on this part.

2.1.3 Variable Transform

In addition to tests, the VnV framework also provides support for pluggable variable transforms that can be used to map injection point parameters into formats that can be consumed by the tests. These test modifiers are developed using the same plugin based C++ pattern used to define the tests and can be included in seperate modifier libraries or as seperate components in existing testing libraries. Figure 5 shows a modifier that extracts a PETSc PC object from a KSP object prior to testing. This allows the developer to call tests that work with the PC object from injection points that only pass the KSP.

2.1.4 The VnV Runtime Module

The VnV runtime module is the driving force behind the framework. This module contains all the functionality required to detect the injection points, parse the configuration file, setup the ADIOS IO engine, load the external testing libraries and run all the tests.

Configuring a simulation to use the VnV framework is a simple, four step task;

- Include the ”vv-runtime.h“ header file.
- Call the VVInit function prior to the first injection point
- Call the VVFinalize function before exiting the main function
- Link the VV library to the execuatble.

The users primary interaction with the VnV runtime module is through the XML configuration file. The full XSD specification for the input files is shown in Figure 6.

A full XML parser was built for the input file specification using XSD2Cpp. The parser allows for reading and writing input files based on the XSD specification and has built in support for input file validation. This makes it incredibly easy to parse and validate the XML input files.

Figure 7 shows a snippet of the XML file required to setup a test at an injection point. The first step to injecting a test is to load the test library. This is as simple as specifying the path to the library in the “testLibrary” element. This element supports the specification of multiple paths, making it very easy to include multiple libraries. In this case, we include a test library located in the tests directory called “testLib.so”.

The second step to injecting a test is to define the injection point. The Phase I prototype can automatically generate an input file that contains all registered injection points; however, including a non-registered injection point is a simple task. The only required parameters for an injection point are the “name” and “markdown” attributes. The “name” attribute represents the unique name given to the injection point in the code. The markdown attribute should be the path to a file that contains the YAML based specification for the injection point (described below).

```
//Transforms allow for the transformation of injection point
//parameters prior to being passed to the testing routines. This
//minimizes the number of tests that need to be written.

//This modifier expects a petsc ksp and returns a petsc PC.
class TestModifier : public IVVTransform {

  PC pc;

  //Decalre the input and output type supported by this Modifier
  std::pair<std::string, std::string> declareExpects() {
     return std::make_pair("KSP","PC");
  }

  // The main transform function
  void* Transform(std::pair<std::string, void*> ip, std::string tp) {
      if ( ip.first.compare("KSP") != 0 || tp != "PC" )
        throw "Parameter types not supported"

      KSP *ksp = (KSP*) ip.second;
      KSPGetPC(*ksp,&pc);
      return (void*) pc;
   }

};

//Boiler Plate code gernerated using the IVVTransform generation
//script.

extern "C" {
  IVVTransform* TestModifier_modifier() {
    return new TestModifier();
  }
};

class TestModifier_proxy {
public:
    TestModifier_proxy(){
    VV::trans_factory["TestModifier"] = Test_Under_Sample_modifier;
  }
};

TestModifier_proxy p;
```

Figure 5: An example of a custom VnV test. In this case we implement a test that checks that the slope and intersection point of the linear function are positive and writes the result to file.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="vv" xmlns:t="vv"
elementFormDefault="qualified">

  <xs:element name= "testLibrary">
    <xs:complexType>
        <xs:sequence>
          <xs:element name="path" type="xs:string" minOccurs="1" maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:complexType name = "testparameter">
    <xs:attribute name = 'from' type = 'xs:string'/>
    <xs:attribute name = 'to' type = 'xs:string'/>
    <xs:attribute name = 'trans' type = 'xs:string' default='default'/>
  </xs:complexType>

  <xs:complexType name = "configparameter">
    <xs:attribute name = 'name' type = 'xs:string'/>
    <xs:attribute name = 'value' type = 'xs:string'/>
  </xs:complexType>

  <xs:element name="testStage">
        <xs:complexType>
      <xs:sequence>
          <xs:element name ="parameter" type="t:testparameter" minOccurs="0" maxOccurs="unbounded" />
      </xs:sequence>
      <xs:attribute name="ipId" type="xs:integer" />
      <xs:attribute name="testId" type="xs:integer" />
    </xs:complexType>
  </xs:element>

  <xs:element name="test">
    <xs:complexType >
      <xs:sequence>
          <xs:element ref="t:testStage" minOccurs="0" maxOccurs="unbounded" />
          <xs:element name ="config" type="t:configparameter" minOccurs="0" maxOccurs="unbounded" />
      </xs:sequence>
      <xs:attribute name="name" type="xs:string" />
      <xs:attribute name="markdown" type="xs:string" />
    </xs:complexType>
  </xs:element>

  <xs:element name="injectionPointStage">
        <xs:complexType>
      <xs:sequence>
          <xs:element name="desc" type="xs:string" maxOccurs="1"  />
          <xs:element name="parameters" type="xs:string" maxOccurs="1"/>
        </xs:sequence>
      <xs:attribute name="stageId" type="xs:integer" />
    </xs:complexType>
  </xs:element>

  <xs:element name="injectionPoint">
        <xs:complexType>
      <xs:sequence>
          <xs:element name="desc" type="xs:string" maxOccurs="1"  />
          <xs:element ref="t:injectionPointStage" minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="t:test" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="name" type="xs:string" />
      <xs:attribute name="markdown" type="xs:string" />
    </xs:complexType>
  </xs:element>

  <xs:element name="scope">
        <xs:complexType>
      <xs:sequence >
          <xs:element ref="t:scope" minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="t:injectionPoint" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="name" type="xs:string" />
    </xs:complexType>
  </xs:element>

  <xs:element name="intro">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="markdown" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="outro">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="markdown" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="exe">
        <xs:complexType>
      <xs:sequence>
          <xs:element name="path" type="xs:string" minOccurs="1" maxOccurs="1" />
          <xs:element ref="t:intro" minOccurs="0" maxOccurs="1"/>
          <xs:element ref="t:outro" minOccurs="0" maxOccurs="1"/>
          <xs:element ref="t:testLibrary" minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="t:scope" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      <xs:attribute name="name" type="xs:string"/>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

Figure 6: The XSD specification for the input configuration file. The VnV framework uses this specification in conjunction with Xsd2Cpp to automatically generate a fully featured XML Parsing library with support for reading, writing and validating XML files.

```
<ns1:testLibrary>
  <ns1:path>../tests/build/testLib.so</ns1:path>
</ns1:testLibrary>

<ns1:injectionPoint name="IP_1" markdown="injection_points.yaml">
 <ns1:test name="LinearTest" markdown="linear_test.yaml">
  <ns1:testStage ipId="-1" testId="1">
   <ns1:parameter from="slope" to="slope"/>
   <ns1:parameter from="intersection" to="intersect"/>
  </ns1:testStage>
 </ns1:test>
</ns1:injectionPoint>
```

Figure 7: An example of the XML configuration file for attaching test to injection points at runtime. In this case, we attach the LinearTest test shown in Figure 4 to the "IP_1" injection point shown in Figure 2.

To specify a test, the user must provide the unique name of the test and the path to the YAML based specification file for the test.

The testStage element allows the user to map the injection point stages to the test stages. This is completed using the ipId and testId parameters. In Figure 7, we attach the LinearTest test shown in Figure 4 to the single stage "IP_1" injection point shown in Figure 2.

Once the input file has been created, turning on testing using the VnV framework is as simple as calling the Initialization function with the correct filename. Runtime validation of the input file is completed at this time. For registered injection points, this includes checks to ensure the parameter mapping specified in the input file are valid. For un-registered injection points, this validation must be completed at runtime. In that case, if a parameter mapping is found to be invalid, the runtime module prints out an error and skips that test.

2.2 Task 2: Prototype Several VnV Tools

Using the framework defined above, the project team developed a small VnV testing library for inspecting the values of data stored in C++ vectors.

Figure 8 shows an example of one of the tests supported in this prototype library. In this case the test calculates the euclidean error between two C++ vectors. The output out the test is a double representing the error.

Other tools included in the library include a tool for calculating vector norms, a tool for asserting that all values in the vector are positive, and a tool for plotting the vector on a line chart. A key goal of the Phase II project will be to extend this library to support evaluations on distributed arrays. Distributed arrays are particularly tricky because the testing algorithms will have no knowledge of how the data is

```
#include "injection.h"
#include <math.h>

class EuclidianError : public IVVTest {

public:


    void decalareParameters(std::map<std::string,std::string> &parameters) {
         parameters.insert(std::make_pair("data0",type));
         parameters.insert(std::make_pair("data1",type));
    }

    TestStatus runTest(adios2::Engine &engine, int testStage,
                       std::vector<double> *data0, std::vector<double> *data1 ) {

       double res = 0.0;
       if ( data0.size () == data1.size() ) {
          for ( int i = 0; i < data0.size(); i++ ) {
             res += powf(data0[i] - data1[i],2);
          }
          engine.Put("error", sqrt(res));
          return SUCCESS;
       }
       return FAILURE;
    }

    static void DeclareIO(adios2::IO &io) {
     io.DefineVariable<double>("error");
    }

    // Autogenerated Boiler Plate code
    ...

};

// Autogenerated Registration code.
...
```

Figure 8: The testing code for the beginnings of a VnV testing library for making statistical assertions regarding the data stored in distributed arrays. Amongst other things, the Phase I prototype supports calculating the Euclidean error between two vectors. The Phase II effort will look to extend this support to include a range of efficient statistical methods that can be applied to data stored in distributed arrays.

distributed across the processors.

2.3 Task 3: Prototype the Graphical User Interface.

The final task of the Phase I project was to develop a GUI for the VnV toolkit. In particular, the Phase I effort focused on developing the software interfaces required to visualize the data output during VnV testing.

After accessing the strengths and weaknesses of multiple different approaches, the project team decided to develop the VnV data visualization interface using HTML and Javascript. The primary benefit of this approach is portability - the report can be displayed in any web browser - but other benefits include interactive components, non-linear data presentation and high levels of customization. Moreover, the server-less nature of the HTML web-page allows for direct publication on any static web hosting service (github.io, AWS S3, etc.).

The goal of the Phase I project was to create an interface that allowed for automated, informative post-processing of the data output during VnV testing. To do this, the project team developed the VnV automated documentation generation system.

This system is built around the fact that simulations are built up of a large number of smaller functions, each with a specific pre-defined task and an expected result. This compartmentalization makes it very easy to write generic templates that describe what is happening at each function and during each injection point. Likewise, while the inputs to VnV tests might change, the overall structure of the outputs is constant. For example, the LinearTest shown above always outputs the slope and intersection points to file. This makes it very easy to write template specifications that can be populated with data during post-processing to produce an informative, customized report.

The VnV framework implements this functionality using a combination of the YAML file format and a custom markdown specification. Figure 9 gives an example of this YAML markdown file that provides templates for the "IP_1" injection point shown in Figure 2 and for the LinearTest test in Figure 4. The fields supported in the YAML specification for each test and injection point are:

- **title:** A descriptive title for the injection point.
- **content:** A markdown formatted string representing the content to be displayed for this injection point in the final report.
- **sections:** A map containing the content for any subsections to be displayed under the original content description. Each subsection is displayed as a collapsible child of its parents content panel and is included in the overall index of the final report.

During compilation, the automatic documentation generation script loops through every injection point recorded in the VnV output file. For each injection point, the script locates the injection point specification using the markdown attribute provided in the input configuration file, renders it in HTML, and adds it as a new section in the final report. Similarly, for each test, the script locates the test specification and renders it as as subsection in the injection point section.

The defining feature of the specification files is the support for writing the content using a custom markdown format that provides support for automating data post-processing and visualization. In particular, this custom markdown format is designed to facilitate direct interaction with the data collected during the VnV testing Phase. In this way, users can write generic markdown specifications that are automatically populated with data during rendering.

The Extended markdown format for numerical simulations, MD-XNS, was developed for the VnV project as a highly customizable system for automatically processing the results obtained from numerical simulations into interactive, informative HTML/JS web pages. The extension itself was built using py-markdown, an open source python library for converting markdown files into HTML. In addition to standard markdown commands, the MD-XNS format supports custom post-processing commands of the form

```
[VV::<name>={...},...],
```

where name is the name of the component the user would like to insert and ... represents a dictionary of configuration options. The key feature of the MD-XNS format is that it supports direct interaction with data stored at each ADIOS step. In the markdown specification, this data is accessed through the following syntax:

```
[DATA::<name>={...}]
```

Here, the name represents the name of the data element to be inserted. The following library allows for specific type formating information, although that support has not been implemented in the Phase I prototype yet. The data that is available in a markdown specification depends on the injection point and test that the specification was written for. For example, the slope, intersect and valid parameters written by the LinearTest test could be queried when writing the test specification for the LinearTest test. In this way, developers of tests can write simple markdown specifications that render into highly informative, interactive data visualizations without ever having to process the data. This allows for the creation of living, breathing VnV reports that evolve as the simulation is developed. Figure 9 shows an example test specification for the "IP_1" injection point and the LinearTest test.

The Phase I prototype has limited support for a number of custom data visualization routines including

- Table: The table command inserts an interactive, sortable and search-able table into the final HTML document. The user can populate the table by entering the information manaully, by providing the name of a csv file, and/or by using data generated at runtime.
- Chart2D: The Chart2D command inserts an interactive Charts.JS chart into the document. The entire array of Charts.JS charts are available through this component, including bar, line, scatter and pie charts. In each case, the chart is configured using a python dictionary entered directly into the markdown.
- VTPView: The VTPView command uses VTK.js to insert interactive 3D visualization of a .VTI files in the final document.
- PostPro: The PostPro function allows the user to set up post-processing scripts for execution during the report generation phase. In this way, users can write simple scripts that parse the data into formats more suitable for use in any of the other data visualization components.
- ThreeJs: The ThreeJS command provides another approach for integrating three dimensional visualization in the final report, in this case using three.js for rendering. This component is particularly useful for viewing meshes.

Figure 10 shows a screenshot of a VnV report generated using this approach. The main layout consists of two components; the index and the content. The index in generated directly from the VnV output file. Each entry in the index represents an injection point that was reached during the execution of the simulation. If a new injection point was reached during the first and last steps of another staged injection point, that injection point is listed as a child node in the index. In this way, the index represents a coarse grained view of the simulations call stack, whereby the injection points represent the nodes. The content

```yaml
IP_1:
  title: Linear Function Evaluation
  content: |
    Evaluating linear functional. The **expected** result
    is value = slope*x + intersection.

    This specification supports all standard markdown
    formating.

LinearTest:
    title: Slope Testing for linear Functions.
    content: |
        The slope for this linear function was [DATA::slope]
        The intersection Point was [DATA::intersect]

        The result from the validation was [DATA::valid] where:
        * 0 indicates it was valid,
        * -1 indicates the slope is less than zero
        * 1 indicates that the intersect was less than or equal to zero

        Here is a plot of the linear function that was tested

        [VV::plot2D={type="Linear", slope=DATA::slope, intersect=DATA::intersect}]
```

Figure 9: An example YAML specification file. In this case, the test LinearTest uses the data API to extract the values for the slope and intersect provided by the injection point. The test specification then uses the support for two dimensional plotting to generate a linear plot with the appropriate slope and intersect point.

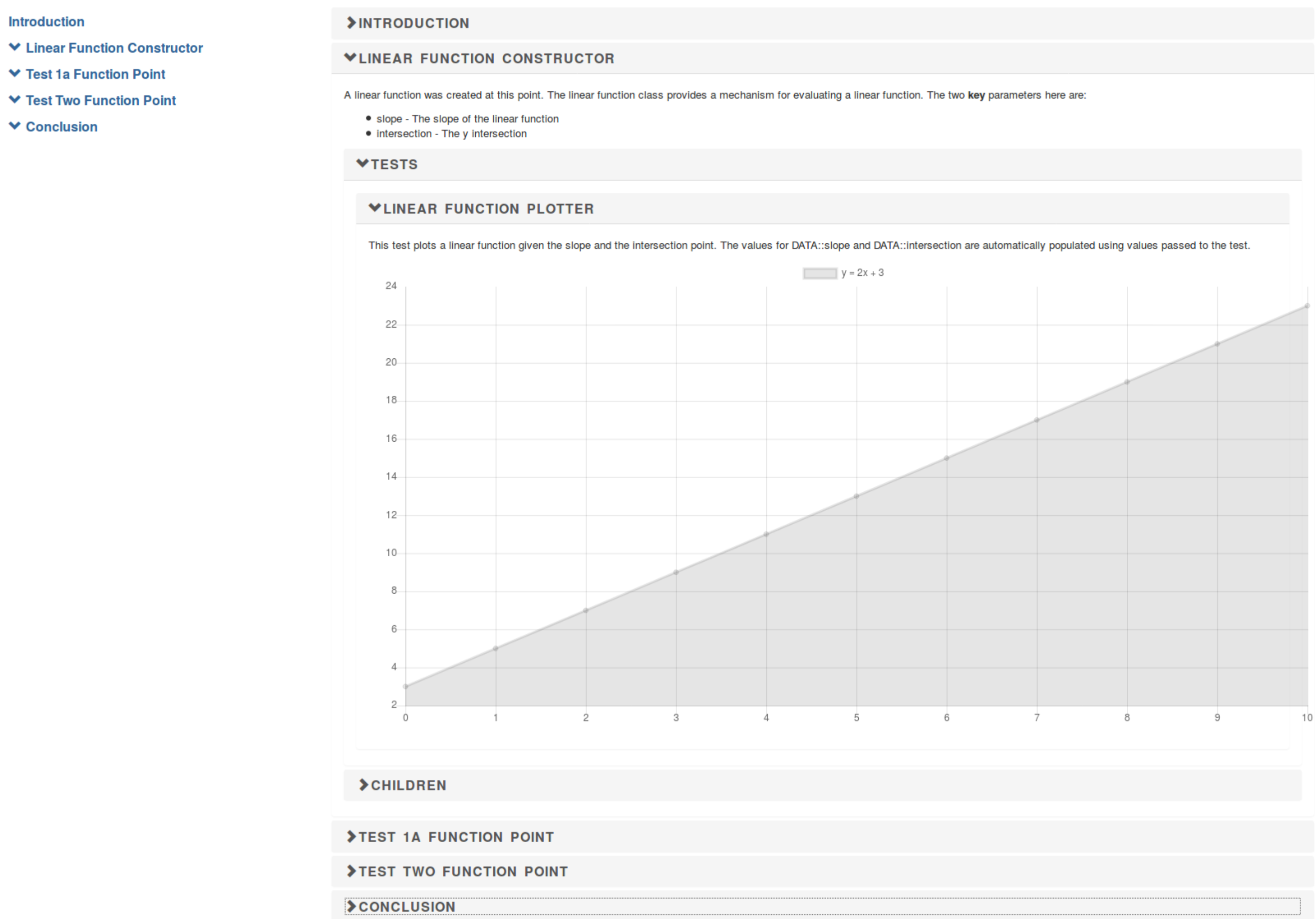

Figure 10: An example of the VnV report generated automatically from the VnV output files and the injection point and test specification files.

section is generated automatically from the YAML specification files. In this case, the user has completed a test titled "Linear Function Plotter" inside an injection point called "Linear Function Constructor".

The key point to note here is that this interface was generated automatically from the VnV output file. Once the test and injection point specifications have been written, no user action is required to build this fully interactive VnV report. As stated earlier, we envision that developers of simulation libraries (e.g., PETSc, libMesh, hypre) would ship their codes with the injection points hard coded into the source code along with any custom VnV testing libraries. Those developers would also provide their users with the injection point specification files, the test specification files, and the input configuration files required to support end-user V&V for there library.

Using these files, the end-users of those packages, i.e., the developers of actual numerical simulations, can then form a single input file that configures V&V testing in every VnV equipt library linked to the final executable. In this way, the V&V toolkit provides a mechanism for generating explainable numerical simulations that not only provide the solution, but also a detailed report on how the solution was obtained and why it can be trusted.

The Phase I proposal had initially stated that the Phase I GUI prototype would include support for setting up input files. However, after developing the framework, and using the input specifications, it became clear that a "click" style user interface would not be all that useful for setting up the configuration files. Rather, text-editor with context aware auto-completions and validations as is present in the NEAMS workbench is likely the best option. As discussed above, the project team developed a XSD specification for the input file format. This specification contains the majority of the information required to setup context aware auto complete when building the input file. However, due to the evolving nature of the inputs; and with the further changes likely to occur in the Phase II project, it would have been highly inefficient to develop the input file GUI support in Phase I. As such, the development of a GUI for editing input files was delayed until Phase II.

The VnV report generation system was written using HTML and JS to ensure portability and applicability for a wide range of users. However, this does not limit, in any way, the ability to view the VnV reports inside the NEAMS workbench. To prove this fact, the project team created a QT application that uses the QWebEngineView to display the HTML VnV report. This QWebEngineView allows for direct interaction with the report in any QT application. In Phase II, the project team will look to improve on this integration by allowing for input files to be specified directly in the MOOSE input files using the NEAMS support for context aware auto-complete and input validation.

3 Summary and Conclusions

During Phase I of the project, RNET have tackled various technical challenges with regard to developing a modern framework for facilitating in-situ end-user verification and validation in general purpose numerical simulation packages. This capability, when fully developed, will provide a convenient interface for creating explainable numerical simulations that not only provide the user with a solution, but also a detailed report on how the solution was obtained and why it should be trusted.

In order to demonstrate feasibility of developing this product, RNET has accomplished the following.

- Developed cross library support for defining and registering injection points in general purpose numerical simulation packages.
- Created an interface for writing and integrating custom V&V tests that can be configured at runtime.
- Developed a custom markdown format that allows for automated post-processing and visualization of testing data.

- Implemented an automated documentation generation script that creates a server-less, interactive VnV report that can be displayed in any web browser.
- Demonstrated how the VnV reports could be viewed in the NEAMS workbench through the QWebEngineView.

The above evaluations demonstrate the technical feasibility of the proposed approach. This platform also has great commercial potential. RNET intends to promote the VnV framework by initially supporting relevant open-source computational tools to leverage their existing customer base.

4 Publications and Presentations

None

References