Project Title: System Verification and Validation Report for SPDFM

S. Shayan Mousavi M.

December 17, 2020

1 Revision History

Date	Version	Notes
Dec 18 2020	1.0	First Draft

Contents

List of Tables

2 Symbols, Abbreviations and Acronyms

symbol	description
T	Test
VnV	verification and validation
SPDFM	Surface Plasmon Dynamics Finite Method
MNPBEM	Metallic NanoParticle Boundary Element Method

The complete table of symbols, abbreviations and acronyms can be found in the SRS document of the software.

This document provides the information on validation and verification plans implemented for the SPDFM software. In this regard, the general approaches and plans are initially discussed and afterwards specific test cases and approaches for validation and verification of functional and nonfunctional requirements (can be found in SRS) are reviewed. VnV plans here are a combination of manual (assigned to a member of the VnV team to assess) and automated testing approaches to evaluate the correctness of the information (whether input or output) or satisfaction of a goal in SPDFM.

3 Functional Requirements Evaluation

3.1 Light Source Calculation Verification (and/or Validation) Tests

Test R 3: Verifying light source setup

1. **Test 1:** Calculation of the electric field of the light source

   Control: Automated

   Initial State: N/A

   Input: light polarity and direction, a frequency, and a meshed geometry are given using input files and input data indicated in Table 1.

   Output: Below outputs should be generated for each of the real and imaginary parts of the electric field separately.

   - Superimposed plot of light wave oscillation towards the light propagation axis (call this line L) calculated by SPDFM and calculated by python built in functions.
   - Difference between two calculated values at each point of the space that is located on the line L. line L is the line that parallel to the direction of the light source and passes the point (0, 0, 0) of the space.
   - Test status: PASS, if the difference mentioned above is below the tolerance-level; FAIL, if the difference is above tolerance-level.

Test Cases:	Test 1	Test 2
Input file Polarity Direction Frequency (THz) Mesh Input	LS_t1.txt 0,1,0 1,0,0 600 G_cube_10node.xml G_cube_10node_physical_region.xml G_cube_10node_facet_region.xml	LS_t2.txt 0,1,0 1,0,0 30000 G_cube_10node.xml G_cube_10node_physical_region.xml G_cube_10node_facet_region.xml
Test Cases:	Test 3	Test 4
Input file Polarity Direction Frequency (THz)	LS_t1.txt 0,1,0 1,0,0 600	LS_t2.txt 0,1,0 1,0,0 30000
Mesh Input	G_cube_20node.xml G_cube_20node_physical_region.xml G_cube_20node_facet_region.xml	G_cube_20node.xml G_cube_20node_physical_region.xml G_cube_20node_facet_region.xml
Test Cases:	Test 5	Test 6
Input file Polarity Direction Frequency (THz)	LS_t1.txt 0,1,0 1,0,0 600	LS_t2.txt 0,1,0 1,0,0 30000
Mesh Input	G_cube_40node.xml G_cube_40node_physical_region.xml G_cube_40node_facet_region.xml	G_cube_40node.xml G_cube_40node_physical_region.xml G_cube_40node_facet_region.xml

Table 1: Input data (files) for automated testing of the light source setup

Test Case Derivation: This test evaluates if SPDFM is correctly calculates both real and imaginary parts of the electric field of the light

Test Cases:	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
Output files	Re_t1.pvd Im_t1.pvd	Re_t2.pvd Im_t2.pvd	Re_t3.pvd Im_t3.pvd	Re_t4.pvd Im_t4.pvd	Re_t5.pvd Im_t5.pvd	Re_t6.pvd Im_t6.pvd

Table 2: Output files for visual inspection of the light source setup

source or not. As testing all the possible polarities, directions, frequencies, and geometries is beyond scope of this work, here only electric field of a few conditions is tested. In this regard, test light source here has 2 sets of light polarity and direct ((1, 0, 0; 0, 1, 0) and (0, 1, 0; 1, 0, 0)), and 2 frequencies (600 THz, 30000 THz) in 2 different meshed geometries (a 40nm x 40 nm cubic mesh, and a core-shell spherical meshed geometry with 40 nm outer diameter and 20 nm inner diameter). In this test, cubic mesh represent the vacuum space and core-shell represents a 20 nm diameter particle in a 40 nm diameter empty volume.

How test will be performed: This test can be executed by running test_ls.py. Shayan Mousavi is responsible for writing and execution of this test.

2. **Test 2:** Visual inspection of the electric field propagation of the light source

   Control: Manual

   Initial State: N/A

   Input: light polarity and direction, a frequency, and a meshed geometry are given using input files and input data indicated in Table 1.

   Output:

   - exported .pvd files containing the interpolated real and imaginary parts of the electric field of the light source in the entire space. These files are named as mentioned in Table 2.

   Test Case Derivation: This test visually evaluates the propagation of electric field in space. In this regard, the extracted pvd map of the electric field should be opened by a pvd reader (suggestion: Paraview software). If 3D colour map view is selected for illustration of the result (which is suggested) alternating domains are expected to be seen with domain width equal to wavelength of the light source. In this regard, for 600 THz source the wavelength is $\sim$499 nm, thus no alternation should be seen in real domain (due to the size of the mesh) and a dipole domain alternation should be observed in the result. The wavelength for the 30000 THz source is $\sim$10 nm, thus, alternating domains of the same width is expected. However, user should observe a $\frac{\pi}{2}$ shift between real and imaginary components. Same colour domains are elongated towards polarity vector and colour alternation should happen towards the propagation vector (direction vector).

How test will be performed: The pvd maps can be obtained by executing test_visual_ls.py. Shayan Mousavi is responsible for writing and execution of this test.

Test R 4: Verifying calculated electric field and electric current density

1. **Test 3:** Plasmon enhanced electric field calculation compared to boundary element simulation

   Control: Automated

   Initial State: N/A

   Input: Input data will be fed to the program through input files listed in Table 3. These files include all the required data for initiation of a FEM simulation.

   Output: Below outputs should be generated:

   - Superimposed plot of electric field intensity vs. distance from the sphere surface for both MNPBEM simulated electric field and SPDFM. These plot will be provided for two directions one passing the centre of space point (0, 0, 0) and is parallel to (0, 1, 0) and the other one is parallel to (1, 0, 0).
   - Difference between two calculated electric field values at each point of the space in plots mentioned above.

   Test Case Derivation: In this test excited electric field by a plane wave of 400 nm wavelength, around a 20 nm diameter sphere is calculated

Test Cases:	Test 1	Test 2
Input files	Input_t1.pvd	Input_t1.pvd
Polarity	0, 1, 0	0, 1, 0
Direction	1, 0, 0	1, 0, 0
Lambda(init) (nm)	400	400
Lambda(fin) (nm)	500	500
Steps	1	1
ε_0 (F/m)	8.85×10^{-12}	8.85×10^{-12}
μ_0 (H/m)	1.25×10^{-6}	1.25×10^{-6}
μ_1 (H/m)	1.25×10^{-6}	1.25×10^{-6}
γ (THz)	17.94	17.94
Plasma freq. (THz)	2165	2165
β^2 (m^2/s^2)	1.16×10^{12}	1.16×10^{12}
Input mesh	G_Shell_t1.xml	G_Fill_t2.xml
	G_Shell_PR_t1.xml	G_Fill_PR_t2.xml
	G_Shell_PR_t1.xml	G_Fill_FC_t2.xml

Table 3: Input data required for SPDFM a complete FEM simulations

by SPDFM and MNPBEM toolbox. MNPBEM is a boundary element method (BEM) software for simulating plasmon activities is nanoparticles (Hohenester and Trügler, 2012). Although in MNPBEM parameter determination is not as flexible as SPDFM and some discrepancies are expected due to the implementation of different theories (local quasi-static vs. non-local hydrodynamic) and different techniques (BEM vs FEM), MNPBEM result can still be compared with FEM simulations that are meshed a boundary instead of the whole volume (G_Shell_t1.xml). In this regard, for a 20 nm diameter sphere made of gold (parameters obtained from Grady et al. (2004)) that is only meshed on the surface, electric field intensity is compaired between MNPBEM and SPDFM simulations in direction of light propagation (1, 0, 0) and polarity (0, 1, 0). To see how presence of the sphere volume affects these results, a 20 nm fully meshed gold sphere is also compared with the MNPBEM results.

How test will be performed: Executing test_Efield_FEM_MNPBEM.py automatically initiates the test and results will be printed. Shayan Mousavi is responsible for writing and execution of this test.

2. **Test 4:** Visual inspection of the plasmon enhanced electric field calculation (SPDFM Vs. MNPBEM)

   Control: Manual Initial State: N/A

   Input: Input data will be fed to the program through input files listed in Table 3. These files include all the required data for initiation of a FEM simulation.

   Output: Colormap plot of the generated electric field calculated by MNPBEM (MNPBEM_20nm_wl_400nm.png) and pvd file showing the result obtained from SPDFM (SPDFM_20nm_wl_400nm.pvd).

   Test Case Derivation: In this test the electric field around a 20 nm diameter sphere is calculated by SPDFM and MNPBEM toolbox. Polarity, direction, and in general, electric field distribution in space for both simulations BEM and FEM simulations should be the same.

   How test will be performed: Executing test_Efield_FEM_MNPBEM.py automatically runs the SPDFM simulation and saves the results in

SPDFM_20nm_wl_400nm.pvd. MNPBEM_20nm_wl_400nm.png is already available in the repository and files can be compared manually by the user. Shayan Mousavi is responsible for writing and execution of this test.

3. **Test 4:** visual inspection of the electric field calculation with respect to the light source

   Control: Automated

   Initial State: N/A

   Input: Input data will be fed to the program through input files listed in the table **??**.

   Output:

   Test Case Derivation: Physically polarity of the light source and the electric response of the system should be the same. Therefore,

   How test will be performed:

4. **Test 14:** Light source electric field evolution

   Control: Manual

   Initial State: **p**, **d**, and wavelength are previously given to the software.
   p=(1,0,0)
   d=(0,1,0)
   wavelength=700 nm
   The location vector, r, and time, t, will be given as below.

   Input: R = $\{\mathbf{r} = (0.1 * r_x, 0, 0) | \forall r_x \in \aleph, r_x \in [0, 10]\}$
   T = $\{t | \forall t \in \aleph, t \in (0, 10]\}$

   Output: for t=0, plot of evolution of electric field by R. for r=(0, 0, 0), plot of evolution of electric field in time.

   Test Case Derivation: In this test $E_i = cos(k\,\mathbf{d}.\mathbf{r} - \omega t) - i\,sin(k\,\mathbf{d}.\mathbf{r} - \omega t)$ is plotted with respect to time and space and the examiner is responsible to see if the behaviour of the function is as expected or not.

   How test will be performed: Using Pytest library in python. Codes can be found in **E_field_plot_test.py** in the src folder. Shayan Mousavi is responsible for execution of this test.

Test R 6: Correctness of the calculated electric field density (E) and electric current density ($\mathbf{J}_{\mathbf{HD}}$)

1. **Test 15:** Proper implementation of equations in code

   Control: Manual

   Initial State: performance of FEniCS toolbox has not been tested.

   Input:N/A

   Output: N/A

   Test Case Derivation: Walkthrough

   How test will be performed: Shayan Mousavi and Dr. Alexander Pofelski are responsible for walkthrough the whole codes after MG and MIS documents developed and verify the functionality of the code and validate that theoretical equations are properly fed to FEniCS PDE solver.

2. **Test 16:** Amplitude of E and J_{HD} in absence of light source

   Control: Manual

   Initial State: performance of FEniCS toolbox has not been tested.

   Input:**p**=(0,0,0)
   d=(0,1,0)
   wavelength=700 nm
   T = $\{t|\forall t \in \aleph, t \in (0, 10]\}$
   materials properties file (.csv): complete_set.csv
   mesh file (.msh): cylinder_3d.msh

   Output: for all the nodes value of E and J_{HD} shall be zero

   Test Case Derivation: As polarity of the incident light is p=(0,0,0), in fact, the amplitude of the incident electric field is zero. Thus, in absence of the excitation source no plasmonic activity should be observed in the medium.

   How test will be performed: Shayan Mousavi is responsible for inputting the data an inspecting if E and J_{HD} values are equal zero.

3. **Test 17:** Amplitude of E and J_{HD} in absence of light source

   Control: Manual

   Initial State: Performance of FEniCS toolbox in python has not been tested.

   Input:**p**=(0,0,0)
   d=(0,1,0)
   wavelength=700 nm
   T = $\{t|\forall t \in \aleph, t \in (0, 10]\}$
   materials properties file (.csv): Material properties for gold (complete_set.csv)
   mesh file (.msh): A 3d meshed cylinder (cylinder_3d.msh)

   Output: for all the nodes value of E and J_{HD} shall be zero

   Test Case Derivation: As polarity of the incident light is p=(0,0,0), in fact, the amplitude of the incident electric field is zero. Thus, in absence of the excitation source no plasmonic activity should be observed in the medium.

   How test will be performed: Shayan Mousavi is responsible for inputting the data an inspecting if E and J_{HD} values are equal zero.

4. **Test 18:** Amplitude of E and J_{HD} in absence of light source

   Control: Manual

   Initial State: performance of FEniCS toolbox in python and MNPBEM toolbox in MATLAB has not been tested.

   Input:**p**=(1,0,0)
   d=(0,1,0)
   wavelength=532 nm
   t=0
   materials properties file (.csv): Material properties for gold complete_set.csv
   mesh file (.msh): A meshed sphere of 12 nm diameter

   Output: .vtk 3D intensity map of the meshed geometry

Test Case Derivation: Similar input shall be fed to the MNPBEM toolbox in MATLAB and the electric field enhancement be extracted for the 3D meshed geometry. As MNPBEM uses boundary element method to calculate a quasi-static approximation of the Maxwell's equations it is not expected to have exactly the same results. However, as both nonlocal hydrodynamic theory and quasi-static theory trying to solve similar equations roughly similar behaviour on the surface of the structure is expected.

How test will be performed: Shayan Mousavi is responsible for executing both SPDFM and MNPBEM simulations. Shayan Mousavi, Dr. Gianluigi Botton, and Dr. Alex Pofelski are responsible for evaluating how these two responses are close.

3.2 Tests for Nonfunctional Requirements

3.2.1 Usability

Test NR1: Capability of execution of the software

1. **Test 19:** Usability

   Type: Usability Survey

   Initial State: The system being used should already have Python3, and FEniCS toolbox installed on.

   Input/Condition: A usability survey with the questions listed in Section 5.2. For execution of a simulation, data provided in Test? is suggested to be used.

   Output/Result: Survey results

   How test will be performed: each participant shall install the software on a system and try to run a simulation. Respondents will be asked to rank their experience of installing and running a module. A final average grade of 3 will indicate that the users found the system to have average usability. The higher the score,the better the perception of usability. Shayan Mousavi and Alexander Pofelski shall be participate this test. This approach is suggested by Michalski (2019).

3.2.2 Maintainability

Test NR2: Maintainability and expandability of the software

1. **Test 20:** Maintainability

   Type: Maintainability Walkthrough

   Initial State: Maintainability of the repository and external toolboxes used in this software such as FEniCS has not been tested.

   Input/Condition: production version of SPDFM has been released.

   Output/Result: A graded report describing the maintainability of the repository

   How test will be performed: Dr. Alexander Pofelski shall check the repository for the following documentation: SRS, VnV Plan, MG, MIS, User Guide. He shall mark 1 point for each of the above documents. He shall read through each of the above documents and provide a grade between 1 and 5 for clarity of the writing. A score of 1 represents a document that is hard to understand, and a score of 5 represents a document that is easy to understand. The user should also grade the traceability of each document. A score of 1 represents no links within the report, and a score of 5 represents many links between sections of the report. The user shall then divide the sum of the scores for all of the reports by 5. A final average grade of 3 will indicate that the users found the system to have average Maintainability. The higher the score,the better the perception of Maintainability. This approach is suggested by Michalski (2019).

3.3 Traceability Between Test Cases and Requirements

Table 4 shows the connection between functional and nonfunctional requirements and the tests provided in this document.

4 Unit Test Description

[Reference your MIS and explain your overall philosophy for test case selection. —SS] [This section should not be filled in until after the MIS has been completed. —SS]

	R1	R2	R3	R4	R5	R6	NR1	NR2
Test1		X						
Test2		X						
Test3		X						
Test4		X						
Test5		X						
Test6		X						
Test7		X						
Test8		X						
Test9				X				
Test10					X			
Test11					X			
Test12	X							
Test13			X					
Test14			X					
Test15						X		
Test16						X		
Test17						X		
Test18						X		
Test19							X	
Test20								X

Table 4: Traceability Matrix Showing the Connections Between Tests and Functional and Nonfunctional System Requirements

The modular design of SPDFM is introduced in MG document (MGS), and discussed in MIS document (MIS); according to these documents SPDFM is consist of eight modules. These modules are assigned to input the data, input the mesh geometry, storing and organizing the data, calculate the electric field and electric current density, and output the data.

4.1 Unit Testing Scope

[What modules are outside of the scope. If there are modules that are developed by someone else, then you would say here if you aren't planning on verifying them. There may also be modules that are part of your software, but have a lower priority for verification than others. If this is the case, explain your rationale for the ranking of module importance. —SS]

In the process of SPDFM verification, the modules that are the most emphasized on are the input (M5 and M6, MG) and output (M8, MG) modules. As the finite element solver module (M7, MG) uses an external finite element solver (FEniCS), and due to the fact that the obtained results from this module are separately verified within system verification section (Section **??**), verifying M7 is beyond the unit testing scope. Moreover, hardware hiding module (M1, MG), SPDFM control module (M2, MG), and data structure module (M3, MG) are also not being tested here. About the data structure module (M3), it should be mentioned that as this module is being used in all other modules, it can be assumed that this module is being tested indirectly while others are verified.

4.2 Tests for Functional Requirements

[Most of the verification will be through automated unit testing. If appropriate specific modules can be verified by a non-testing based technique. That can also be documented in this section. —SS]

4.2.1 Module 4: Constant parameters module (M4)

[Include a blurb here to explain why the subsections below cover the module. References to the MIS would be good. You will want tests from a black box perspective and from a white box perspective. Explain to the reader how the tests were selected. —SS]

As the constant parameter module (M4) is assigned to be a template object that holds the constants used in project, unit testing is performed by checking all the values.

1. **Test**

   Type: Automatic

Test Cases:	Test 1	Test 2	Test 3	Test 4
Input files	Input_t1.pvd	Input_t2.pvd	Input_t3.pvd	Input_t4.pvd
	Im_t1.pvd	Im_t2.pvd	Im_t3.pvd	Im_t4.pvd

Table 5: Input data (files) for input module (M5) unit testing

Initial State: N/A

Input: The constant values used in the SPDFM are stored in constants.txt.

Output: PASS, if all the constant values in the template module is equal to (in the tolerance level distance of) the values stores in the constant.txt; FAIL, otherwise.

Test Case Derivation: The constant parameters used in the code should be equal to the constant parameters in Table 2 of the SRS document.

How test will be performed: By execution of test_const.py all the values in the constant parameter object (template module M4, MG) will be tested.

4.2.2 Module 5: Input parameters modules (M5)

1. **Test**

   Type: Automatic

   Initial State: N/A

   Input: In this test the input data is as stated in Table 5. constants.txt.

   Output: Pass, if all the constant values in the template module is equal to (in the tolerance level distance of) the values stores in the constant.txt; Fail, otherwise.

   Test Case Derivation: The constant parameters used in the code should be equal to the constant parameters in Table 2 of the SRS document.

   How test will be performed: By execution of test_const.py all the values in the constant parameter object (template module M4, MG) will be tested.

2. test-id2

   Type: [Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic —SS]

   Initial State:

   Input:

   Output: [The expected result for the given inputs —SS]

   Test Case Derivation: [Justify the expected value given in the Output field —SS]

   How test will be performed:

3. ...

4.3 Tests for Nonfunctional Requirements

[If there is a module that needs to be independently assessed for performance, those test cases can go here. In some projects, planning for nonfunctional tests of units will not be that relevant. —SS]

[These tests may involve collecting performance data from previously mentioned functional tests. —SS]

4.3.1 Module ?

1. test-id1

   Type: [Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic —SS]

   Initial State:

   Input/Condition:

   Output/Result:

   How test will be performed:

2. test-id2

Type: Functional, Dynamic, Manual, Static etc.

Initial State:

Input:

Output:

How test will be performed:

4.3.2 Module ?

...

4.4 Traceability Between Test Cases and Modules

[Provide evidence that all of the modules have been considered. —SS]

References

Spdfm/docs/design/mg at master · shmouses/spdfm. https://github.com/shmouses/SPDFM/tree/master/docs/Design/MG. (Accessed on 12/16/2020).

Spdfm/docs/design/mis at master · shmouses/spdfm. https://github.com/shmouses/SPDFM/tree/master/docs/Design/MIS. (Accessed on 12/16/2020).

Nathaniel K Grady, Naomi J Halas, and Peter Nordlander. Influence of dielectric function properties on the optical response of plasmon resonant metallic nanoparticles. *Chemical Physics Letters*, 399(1-3):167–171, 2004.

Kirankumar R Hiremath, Lin Zschiedrich, and Frank Schmidt. Numerical solution of nonlocal hydrodynamic drude model for arbitrary shaped nano-plasmonic structures using nédélec finite elements. *Journal of Computational Physics*, 231(17):5890–5896, 2012.

Ulrich Hohenester and Andreas Trügler. Mnpbem–a matlab toolbox for the simulation of plasmonic nanoparticles. *Computer Physics Communications*, 183(2):370–381, 2012.

Stefan Alexander Maier. *Plasmonics: fundamentals and applications*. Springer Science & Business Media, 2007.

Peter Michalski. Latticeboltzmannsolvers/docs/vnvplan/systvnvplan at master · peter-michalski/latticeboltzmannsolvers. https://github.com/peter-michalski/LatticeBoltzmannSolvers/tree/master/docs/VnVPlan/SystVnVPlan, 2019. (Accessed on 10/29/2020).

mit.edu. Checklist for code walkthroughs (draft, version 1.2, 10/30/97). http://www.mit.edu/~mbarker/ideas/checkcode.html, 1997. (Accessed on 12/14/2020).

Peter Monk et al. *Finite element methods for Maxwell's equations*. Oxford University Press, 2003.

S. Shayan Mousavi. Spdfm/docs/problemstatement at master · shmouses/spdfm · github. https://github.com/shmouses/SPDFM/tree/master/docs/ProblemStatement, 2020a. (Accessed on 10/29/2020).

S. Shayan Mousavi. Spdfm/docs/srs at master · shmouses/spdfm · github. https://github.com/shmouses/SPDFM/tree/master/docs/SRS, 2020b. (Accessed on 10/29/2020).

Spencer Smith. Blankprojecttemplate/docs/srs/srs-checklist.pdf · master · w. spencer smith / cas741 · gitlab. https://gitlab.cas.mcmaster.ca/smiths/cas741/-/blob/master/BlankProjectTemplate/docs/SRS/SRS-Checklist.pdf, Sept 2020. (Accessed on 10/29/2020).

5 Appendix

This is where you can place additional information.

5.1 Symbolic Parameters

The definition of the test cases will call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.

5.2 Usability Survey Questions?

Using the following rubric please rate the five statements found below (this rubric is suggested by Michalski (2019)):

1. The formatting of the input file was easy to understand.

1 - strongly disagree
2 - somewhat disagree
3 - neither agree nor disagree
4 - somewhat agree
5 - strongly agree

2. The location to place the input file was easy to find.

1 - strongly disagree
2 - somewhat disagree
3 - neither agree nor disagree
4 - somewhat agree
5 - strongly agree

3. Navigating to the correct module was straightforward.

1 - strongly disagree
2 - somewhat disagree
3 - neither agree nor disagree
4 - somewhat agree
5 - strongly agree

4. The MG and MIS of this product explain the modules well.

1 - strongly disagree
2 - somewhat disagree
3 - neither agree nor disagree

4 - somewhat agree
5 - strongly agree

5. I would recommend this product.

1 - strongly disagree
2 - somewhat disagree
3 - neither agree nor disagree
4 - somewhat agree
5 - strongly agree