DARPA ZENITH V1 CODE README FILES

By

Tomu Hisakado, Devin Vollmer, Alicia Dautt-Silva, Alexander Laut, Paul Schroeder

Report prepared for DARPA Zenith under award number FA2377-24-C-B009.

This is an informal progress report prepared for the contracting agency. The results and data may be preliminary or tentative and therefore are subject to revision or correction. This report may contain patentable material on which patent applications have not yet been filed and further distribution of this report should not be made without the prior approval of the contracting agency or the contractor.

GENERAL ATOMICS PROJECT A30682
JULY 2024

GENERAL ATOMICS

DARPA Zenith V1 ReadMe Files

Combined ReadMe files

1. Physical Modeling

This folder contains the files to simulate the surface form for a mercury and air interface using COMSOL aw well as additional files and code for processing files in Matlab and Excel. Please note Code V0 Drop material is also included with no required revisions. Additional new items are mentioned below.

1.1. Code Version 1:

FILE: 0.5m Parabola Centered and Symm 4mm equal_4_21_24.mph

FILE: COMSOL_3Dver11.m

Description: The segmented version of the liquid mirror is simulated to 1.0 seconds with numerous 2mm Mercury Reflective Section (MRS). 2mm wide MRSs provide the lowest peak-to-valley (PV) and are separated by 2mm. The file starts with a 2mm air gap separation in order to tune the widths of each MRS to 3mm and 3.75mm and observe the resulting PV. Electrowetting is applied to each MRS with mercury and air interface. No electrostatic forces were added as the COMSOL interface is not suited to deterministically add 200 syntax codes for each liquid/wall interface through a one line only interface.

1. Go to Study and click COMPUTE. The run time is 2hrs 4min.
2. Go to Results and 1D Plot Group 7: Each lineout of the mercury and air interface is selected for export.
3. Right click on Line Graph 1 under 1D Plot Group 7 and click on "Add plot data to export".
4. In the Results section under Export: Plot XX will appear. Select File type .txt and Filename and EXPORT.
5. Open the file and manually remove headers and keep the two data columns.
6. The exported COMSOL lineout contains all sequence frames within a 1 second time interval. After opening the file in EXCEL, copy and paste the last frame and save as .csv file in order to observe the PV of one time sequence. The last time frame or first time frame contains about 3800 rows that describe the surface form in 2D.
7. Open MATLAB FILE: COMSOL_3Dver11.m and run. A residual surface error plot will be generated with newly fitted coefficients for optical sag of the parabola.

   Note: The interpolated 3D surface will overshoot the true PV values. In order to observe the residual surface map quickly, it is recommended to use 4D Technologies, 4Sight software. There are steps involved in the transfer process from COMSOL to Matlab to 4Sight (if 4Sight is available to the general public), but it does aid in an accurate and fast analysis.
8. Tuning of the width of each MRS is done by: Component 1-Geometry 1- Rectangle 1,2,3,4... and changing the width from 2mm to 3mm for all MRS.
9. To change the slewing rate please see accompanying word document.

10. To turn off gravity go to Laminar Flow-> Gravity-> Input for gravity the syntax for x and y as zero.

11. Changing the focal length is also possible.

    Note: The parabolic fitted equation has been updated: zz=0.7503 + 0.0003334*xtest.^2 + 0.0003334*ytest.^2;

File outputs: 0.5m Full Parabola 4mm equal slew 1 deg per sec BEGIN.csv are provided from the .mph file.

FILE: High Resolution positioning 03_31_24 revised 04_20_24.xlsx

Description: The position and angles have been updated and how to use the excel sheet is explained in the file:

FILE: PS_x10 visc 800V 10um 01_31_24

Description: The continuous Hg film has an applied voltage of 800V across the top surface in a 2D axisymmetric model. There is a volumetric force applied across mercury that uses two forces: an electrostatic force of attraction and frictional surface force. Lastly, a voltage controlled contact angle determined by the YL equation. In order to bound the problem, the viscosity is increased by a factor of 10 in order to overcome the roadblock of long fluid oscillation times.

1. Go to Study 1 and click COMPUTE. The run time is about 35 minutes.

2. Scroll down to Results-> Select: 2D Plot Group 4 -> Surface 1

3. For surface 1, Click on green/red arrow to select the Physical Quantity of interest. Ex: Electric potential (V).

4. Select Animate above.

   Note: The initial geometric boundary conditions are sensitive so the voltages selected and volume used have to be near the final equilibrium geometry or else the simulation will not converge. This is presumably due to the fast changes of the liquid to air interface which produce a nonlinear effect.

1.2. Code Ver 0:

COMSOL files simulate a 2D lineout for a mercury surface in a multi-tub or container configuration. The file incorporates slewing and other adjustable parameters as described in the word file:COMSOL Multiphysics Simulations for the liquid metal surface 04_01_24.docx. The post analysis uses the Matlab file: COMSOL_3Dver7.m to generate figures and a dataset for the residual surface form map based on the simulated surface form in 2D done in COMSOL. The excel file is a basic spread sheet that estimates the angle needed for each tub. The file name is: High Resolution positioning 03_31_24.xlsx.

Demonstration of the Mercury-Air interface with multi containers in a curved geometry with slewing using COMSOL is provided in the file.

FILE: 2D Flat x 11 2mm Hg and Air w ACDC Pulse FULLY ON tilted at 10 deg NO Fes Z_ OSCILLA 2sec w SLEWING _PARAB ALIGNED Center tub 0 deg OFF 4_01_24.mph

FILE: COMSOL_3Dver7.m

1. A word file is provided to describe additional details of the .mph file and export process of the liquid Mercury surface in 2D.
2. The .m file for Matlab is used for post analysis and rotates the 2D lineout and provides an interpolated surface and residual surface error.

Demonstration of the mercury-air interface with a thin dielectric side walls made of quartz and electrostatic surface forces using COMSOL is provided in the file.

FILE: 2D Flat 20mm Hg e is 80 and Air AND DIELECTRIC w ACDC Pulse FULLY ON _ NEW BC ON RHS _ ADDED ON Fes RHS BOT TOP 4_01_24 CORRECTED Left Side and FORCE_WORKS 200ms.mph

3. Go to Study 1 and click COMPUTE. The run time is about 1 hr and 8 minutes.
4. Scroll down to Results-> Select: 2D Plot Group 3 -> Surface 1
5. For surface 1, click on green/red arrow to select the Physical Quantity of interest. Ex: Electric Potential.
6. Select Animate above to simulate the electric potential fields over time.
7. Note: For a metal conductor the vendor suggests setting the dielectric constant to unity for a metal. The value has been set arbitrarily to 80. Metals should have a dielectric constant of infinity so further tuning could be done to observe no field lines in the metal conductor.

Demonstration of Adhesion between the charge liquid metal and surface interface using COMSOL was performed systematically in the following file.

FILE: x10 WORKs TOP SURFACE FORCE ONLY _ALL SURFACES Inverted TALLER 03_04_24 FINAL CONFIGURATION Floor at 800V surf charge on bottom 02_18_2024.mph

1. Go to Study 1 and click COMPUTE. The run time is about 47 minutes.
2. Scroll down to Results-> Select: 2D Plot Group 4 -> Surface 1
3. For surface 1, Click on green/red arrow to select the Physical Quantity of interest. Ex: Volume Force z-dir.
4. Select Animate above.

Note: Steps for providing a partition or half ellipse shaped mercury film are listed in the Geometry Tab.

The COMSOL files are related to the Continuous Film of Mercury with an applied voltage on an incline.

Demonstration of electrostatic forces only on the top surface of mercury using voltage/distance with a dielectric on the floor was performed. The .mph file assumes a frictional forces between the fluid and solid interface with a coefficient of 0.1 or 5.5 degrees.

FILE: FINAL CONFIGURATION Floor at 400V surf charge on bottom 02_18_2024.mph

Note: The threshold of slippage was between 400V and 800V once a domain specified dielectric was integrated.

Note: The electrostatic repulsion forces are not integrated between the top and bottom charged surfaces as the surfaces deviate away from a parallel plate configuration.

Note: The electrical potential does not integrate an Infinite Element Domain to specify the potential at infinite distances away to be zero in a mm scale simulation.

Consultation with COMSOL seems to indicate that it is an unknown if CFD and AC/DC can be used simultaneously with the Infinite Element Domain feature.

1. Go to Study 1 and click COMPUTE. The run time is about 47 minutes.
2. Scroll down to Results-> Select: 2D Plot Group 4 -> Surface 1
3. For surface 1, click on green/red arrow to select the Physical Quantity of interest. Ex: Volume Force r-dir.
4. Select Animate above.

2. Mechanical Modeling

2.1. Changes from V0

The included the phase unwrapping .py script is intended for simulating piston modes. This code produced a .mat that would be used to test the capability of a Gerchberg-Saxton Algorithm.

2.2. Notes

This folder contains the ANSYS 2021 R2 Project for the 12 arm Multi-mode Deformable Mirror (MDM) dish used for finite element analysis (FEA), and the post processing codes used for subsequent analysis. The ANSYS simulation is used to model strength of materials and actuator influence functions of the dish while under inertial loads. ANSYS projects are sensitive to subfolder renaming and relocation, so do not modify the folder structure unless required.

2.3. Exporting data from ANSYS

Unit system; ALWAYS USE MILLIMETERS for these prep/post analysis steps.

When exporting ANSYS nodal results, ensure you are exporting results from a surface only (not a volume).

Users will have to navigate to specific load steps or configure displacements as needed to produce desired FEA results for influence functions, inertial loads, etc. The project was saved in a state that enables users to readily extract these results and also includes a project archive in case they wish to revert to the original model.

a) Open the "Influence Function" system in ANSYS Workbench

b) Select the displacement results for the mirror's top parabolic surface

c) Right click the result and "Export"

d) Save .txt results in the "GA Zenith Mechanical Modeling Outputs" subfolder.

2.4. Results Outputs

The "GA Zenith Mechanical Modeling Outputs" Subfolder contains various .txt, .py, .dat, and .mat files.

.txt files are nodal results exported from ANSYS in mesh coordinate x, y, z, USUM format. The .py scripts will process the .txt file into .mat or .dat files for use in subsequent analysis. This is required because meshes nodal results are not a regular grid so the .py scripts interpolate the FE nodal data and save it as regular grid data to be used in Matlab and Zemax.

2.5. Post-influence function retrieval

Extract influence functions by running the "ANSYS_nodes_to_grid_031424.py" script. Modify the script to read the latest .txt nodal result file or defined a user-specified results file.

The analysis will have to be rerun re-run for every influence function, so name each .txt file accordingly. Examples are already included in the folder.

The script produces a .mat file of the influence function, and will name them based on the .txt files name.

Important: Using influence functions to construct a complete matrix for the actuator system is described in the other "READ ME" found within "Controls". This is ultimately used to run a Least Square Error Analysis (LSEA) using matrix math.

After generating .mat files for the center actuator, 1 inner radius actuator, and 1 outer radius actuator, transfer these .mat files to the respective "Controls" folder. Remaining influence functins are generated using a rotation matrix transformation.

2.6. Post-Zemax ray-trace analysis

Zemax's grid-sag analysis uses a .dat file to interpolate a sag onto an ideal optical surface.

Run "ANSYS_nodes_to_DAT_031424.py" to convert a nodal displacement result (sag) into a .dat file that's ready to import.

The .dat is ready to import into Zemax.

2.7. Others

There are other .py codes found within. These generate non-Zernike noise parameters that users may find interesting.

3. Controls Design

The code for the Control files was created in MATLAB (R2021b) Update 7. The code is in mlx format for the Live Editor. Each folder has its own readme_xxx.pdf, which is an export of the mlx.

There are 2 folders in the Controls files:

3.1. Gerchberg-Saxton (GS) algorithm (GSalgo folder)

The Gerchberg_Saxton_algorithm.mlx solves the GS algo for one simulated surface image (Zernike polynomial 6th order). The files for this are:

- Gerchberg_Saxton_algorithm.mlx
- mksprecon.m
- sprecon.m
- spunwrap.m
- zernike6th.mat

To run this code, run the mlx file.

3.2. Simulation (Simulation folder)

A Simulation of the two actuators working for this project is created in Simulink. The Constants_for_Simulation.mlx file defines the constants for the motor and PZA, creates the transfer functions, and PID controller. The code generates the closed-loop transfer functions for the actuators and for the dual architecture while showing a brief analysis of the Step-response for each. The code executes the simulation and loads the results of the signals in three different scopes. The files for this are:

- Constants_for_Simulation.mlx
- DualFdbk_PZAwH_MotorDC.slx

To run this code, run the mlx file.

4. Experimental Software

This software suite was developed for control and data acquisition for the DARPA Zenith experimental setup at General Atomics. It is written in Python and relies on a ThorLabs BSC201 One-Channel Benchtop Stepper Motor for angle control, an MCC DAQ for measuring voltage, and a Basler camera for image acquisition. The Python packages for communicating to these various devices are required for this software suite to function.

Examples of code structures that were used to take datasets have been placed in the "example.py".

Below are high level descriptions of commonly used commands.

4.1. LMTpy.Instruments()

Initiates the Instruments class and connects to all the instruments

Example usage:

```
import LMTpy as lp

inst = lp.Instruments()
```

4.2. LMTpy.Experiment()

Initiates the Experiment class and optionally sets the save path for data acquisition

Allows access to more directly control instruments, or allows use of already created commands to automate data acquisition.

-keys:

inst: Instruments class object. If none are specified, the script will look for instruments to connect to by default

FILEPATH: path to the folder you want to save your data to. If not specified, a new folder will be made in the current working directory named the current date and time

Example usage:

```
import LMTpy as lp

exp = lp.Experiment(inst=inst,FILEPATH=path)
```

4.3. Commands in Experiment Class:

4.3.1. jog_motor_timed()

Moves stage position and repeatedly takes images over a set amount of time

Allows user to input direction of step between measurements

Data is saved as a pickle file and includes current voltage reading

-keys:

STEP_SIZE: number of degrees to move per step. Default = 0.5 degrees

DURATION: how long to spend taking images at each step. Default = 10 seconds

START_DELAY: how long to wait at new position before beginning to take images. Default = 5 seconds

IMG_DELAY: time between taking two consecutive images. Default = 0.5 seconds

FOLDER_NAME: folder name to save data to. If not provided, default will create a new folder named the current date and time

SHOW_IMGS: option to display the images as they are taken. Default = False

4.3.2. timed_measurement()

Take a series of images at a single location

Data is returned as a dictionary that the user can then save

keys:

DURATION: how long to spend taking images at each step. Default = 10 seconds

START_DELAY: how long to wait at new position before beginning to take images. Default = 5 seconds

IMG_DELAY: time between taking two consecutive images. Default = 0.5 seconds

SHOW_IMGS: option to display the images as they are taken. Default = False

4.3.3. motor_scan_timed()

Automatically scan through a list of angles, taking a series of images at each position

Data is saved, but can also be returned as a dictionary if a variable is assigned

example: > data = exp.motor_scan_timed(DEGREES=degrees)

keys:

DEGREES: list of specific angles to take measurements at.

DURATION: how long to spend taking images at each step. Default = 10 seconds

START_DELAY: how long to wait at new position before beginning to take images. Default = 5 seconds

IMG_DELAY: time between taking two consecutive images. Default = 0.5 seconds

FOLDER_NAME: folder name to save data to. If not provided, default will create a new folder named the current date and time

SHOW_IMGS: option to display the images as they are taken. Default = False

PAUSE_BETWEEN: option to delay data acquisition at a new position until the user inputs a button to begin

4.3.4. motor.home()

-Moves the motor to the home position to recalibrate it for angle measurements

4.3.5. motor.move()

Moves the motor to a specified position in mm

-keys:

pos_mm: Position within the motor's range of movement, in millimeters from home (0mm)

4.3.6. deg_to_pos()

Calibrated conversion from the desired number of degrees to the correlating motor position

Used within the motor.move() command to move to a specific stage angle

example: > exp.motor.move(exp.deg_to_pos(5)) #moves stage to be tilted 5 degrees

keys:

deg = Number of degrees to convert to motor position

4.3.7. daq.voltage()

Returns the current voltage measurements

4.4. QuickBasler.get_img()

Takes an image using a Basler camera

keys:

camera: ID of camera to take image from

convert_to_rgb: Option to convert the image taken to an rgb color image. Default = True

example usage:

```
import QuickBasler as qb
img = qb.get_img(exp.camera)
```


GENERAL ATOMICS