VISA Validation Suite
USER MANUAL

Joo Thiam Goh
Department of Ocean Engineering
Massachusetts Institute of Technology

Defence Science Organisation
20 Science Park Drive
Singapore 118230
Republic of Singapore

January 13, 1997

Contents

1 Testing the Installation

1.1 Installing the Benchmarks

The VISA distribution comes with a complete set of test problems and reference solutions. To install this set of scripts, follow the few simple steps :

- In the `benchmarks` directory, type

  `make yesperl`

  or

  `make noperl`

  depending on whether your particular system has PERL installed.

- Next you will need to compile the `pltsplit` program. Change to the `./src` directory and type

  `make`

  Then move the executable `pltsplit` to somewhere in your `PATH` .

- If you are using the perl-based scripts, you will also need to move `./bin/plpsub.pl` to a location in your `PATH` .

1.2 Directory Tree

The scripts `MUST` be run from `benchmarks` . They assume that files are stored in the following manner :

```
./input              -> VISA input files
./logs               -> logs of each run
./refs/PE            -> reference solution for PE
./refs/BEM           -> reference solution for BEM
./refs/CORE          -> reference solution for CORE
./refs/COUPLE        -> reference solution for COUPLE
./refs/OASES         -> reference solution for OASES
./bin                -> All the relevant plotting scripts
./bin/YES_perl       -> Use these scripts if perl is present
./bin/NO_perl        -> Use these scripts if perl is not present
./tloss              -> Final combined PLT/PLP files
./tloss/default_plps -> PLP files for NO_perl scripts
```

```
./tloss/stubs        -> PLP template files for YES_perl scripts
./run                -> Results from VISA runs
./src                -> Source for pltsplit program
```

The VISA data files are located in the `./input` directory. Since these represent the standard test set, DO NOT change the environmental variables in these files, as well as the number of field options. You may change the sampling parameters but if you did not install the perl-based scripts, then this might require that you change the combined PLP files in `./tloss` after each run (see below).

1.3 Running the scripts automatically

To run a particular test simply call the appropriate script. For example to run Mike Collins Case D,

`do_mikeExD`

The final combined PLP/PLT files can be found in `./tloss/mikeExD.pl?` . To run *ALL* the tests do

`doALL`

These tests are run SEQUENTIALLY, i.e. one at a time to avoid starting multiple copies of VISA!! The `doALL` script takes an argument `Q` or `q` to indicate `Quick` runs, i.e. when invoked as

`doALL Q` or `doALL q`

the script skip runs that involved more than 3 sectors.

Upon completion, the results of individual VISA runs are stored in `./run` . The directory `./tloss` holds the combined PLP/PLT files, i.e. they contain the VISA solutions as well as any other 'correct' solutions for the problem. The plots are always ordered as OASES, BEM, VISA and CORE. Note that we may not have all the solutions from the various codes for each test case. Table 1 shows the available reference solutions for each of the different test case.

The above scripts execute the entire run, split and packaging for you. The next section describes the individual steps involved. Look at `do_mikeExD` to get an idea of it.

Note that the basename for each test is used consistently for the input data file, scripts and combined PLP/PLT files. For example, `do_mikeExD` , `mikeExD.dat` , `./bin/mikeExD`, `./tloss/mikeExD.pl*`, `./logs/mikeExD.out`, `./run/mikeExD.pl*` forms the complete set of files for running the Mike Collin Case D problem. Here, the basename is `mikeExD` . TABLE 1 contains a short description of the test cases.

1.4 Running the scripts manually

When VISA is done, you will need to split the PLT file, say `X.plt` into separate parts so that they can be merged with the reference solutions in `./refs/*` into a single PLT file. The program that does this is `pltsplit` . The `do_*` scripts calls `pltsplit` to do this for you automatically. The program `pltsplit` can be used in various manners.

- Interactively - simply call `pltsplit` and answer the questions.
- Via command line arguments - this is useful for automating the split in shell scripts (see `do_*`). The order of the input parameters is CRUCIAL.
- Type `pltsplit` -h to see the format of the command line inputs.

`pltsplit` splits BEM, VISA, SAFARI or CORE files into individual traces depending on the number of parameters (N,V,H etc), number of receiver depths and whether we have the forward, backward or total field components. It uses a very simple line counting algorithm and then split the file up based on the information you provided - robustness is questionable but so far it works. It then creates a series of files with unique filenames. For example, if you have N,V,H and 2 receivers with forward, backward and total field components, the file `X.plt` will be split into

```
X.rdo.par1.for.rx1.dat    X.rdo.par1.bac.rx1.dat
X.rdo.par2.for.rx1.dat    X.rdo.par2.bac.rx1.dat
X.rdo.par3.for.rx1.dat    X.rdo.par3.bac.rx1.dat
X.rdo.par1.for.rx2.dat    X.rdo.par1.bac.rx2.dat
X.rdo.par2.for.rx2.dat    X.rdo.par2.bac.rx2.dat
X.rdo.par3.for.rx2.dat    X.rdo.par3.bac.rx2.dat

X.rdo.par1.tot.rx1.dat
X.rdo.par2.tot.rx1.dat
X.rdo.par3.tot.rx1.dat
X.rdo.par1.tot.rx2.dat
X.rdo.par2.tot.rx2.dat
X.rdo.par3.tot.rx2.dat
```

The extensions `for, bac, tot` stands for forward, backward and total field components. The extensions `rdo, gem, saf` stands for VISA, CORE and SAFARI files.

Input file (`.dat`)	**DESCRIPTION**	**REFERENCE**
RSWT2a	Reverb. + Scat. Workshop Test case 2a	COUPLE, CORE
RSWT2b	Reverb. + Scat. Workshop Test case 2b	COUPLE, CORE
RSWT3a	Reverb. + Scat. Workshop Test case 3a	COUPLE, CORE
RSWT3b	Reverb. + Scat. Workshop Test case 3b	COUPLE, CORE
ASAwedgeE	ASA Elastic wedge problem	PE, CORE
torusF	Embedded Torus (FLUID)	BEM, CORE
torusE	Embedded Torus (ELASTIC)	BEM, CORE
gilbert	Gilbert-Evans Fluid	CORE
gilbertE	Gilbert-Evans Elastic	CORE
HorizForce	E/E Test - Horizontal Point force.	BEM, CORE
VertForce	E/E Test - Vertical Point force.	BEM, CORE
PntSrc	E/E Test - Point source	BEM, CORE
TranspHF	Transparent Elastic - Horizontal Force	BEM, CORE
TranspVF	Transparent Elastic - Vertical Force	BEM, CORE
mikeExD	Mike Collins Case D	BEM, CORE
mikeExE	Mike Collins Case E	BEM, CORE
30degLBeam	30-deg Llyod mirror beam F-S CORNER	BEM, CORE
45degLBeam	45-deg Llyod mirror beam F-S CORNER	BEM, CORE
SLeeHC	Single-layer High-contrast E/E test	BEM, CORE
SLeet	Single-layer Transparent E/E test	BEM, CORE
SLeed	Single-layer E/E test	BEM, CORE
SLef	Single-layer E/F test	BEM, CORE
SLfe	Single-layer F/E test	BEM, CORE
norda3a	Modified NORDA 3a - Isovelocity case	OASES, CORE
norda3a25c2	Modified NORDA 3a - c^2 case	OASES
fnuw	Single-layer Fluid transparent problem	OASES, CORE
SACL_circle_F	SACLANT semi-circle ridge (Fluid)	COUPLE, CORE
SACL_step_F	SACLANT step discontinuity (Fluid)	COUPLE, CORE

Table 1: One-line description for the test cases

BEM is treated as SAFARI. The extensions `par1, par2, par3` refers to N, V and H. Finally, the extensions `rx*`, refer to the receiver depths, with the shallow receiver designated first.

WARNING : To properly use the scripts of course assume that you always run with the same set of parameters and receiver depths. Also NO INTEGRAND PLOTS are expected. Table 2 shows the expected OPTIONS and receiver depths for the various test cases.

After the PLT file has been split using `pltsplit` , they can then be combined using the scripts in `./bin`

AGAIN this assumes that the test case has been run using certain wavenumber integration parameters that results in a PLT plot of certain size. If this has been changed, you will need to change the combined PLP file in `./tloss` (unless you are using the PERL-based scripts). TABLE 2 shows the default PLT file size for the VISA runs. Note that the scripts, input data file and the combined PLP files all have the same basename.

We have included PERL-based scripts that automatically extract the PLP information from the VISA runs and merge them into the combined PLP files. This bypassed the above step of manually changing the combined PLP files whenever the PLT size is different. However it means that PERL must be installed on your system. If PERL is not available, simply install the alternative set of scripts in `./bin/NO_perl` . See the section on `Installing the Benchmarks` for more information.

These scripts call `mplot` by default and has a silent mode in which the combined PLT is created but `mplot` is not called. The silent mode (call scripts with argument `S`) is used in the `do_*` script to automatically create the combined PLT file in `./tloss` after the run is completed. The default mode is useful for single runs. The scripts also saved the previous combined PLP/PLT files with an extension `.old` . This allows you to recover the old results if the new ones are worse! (Very useful)

NOTE : These scripts assume that `tcsh` is located in `/usr/local/bin` . (see the first line of each script). If you are unable to run the scripts, check to make sure that `tcsh` is where it should be. If not simply create a link to it (you will need to be `root` to do this) ,

```
ln -sf wherever_tcsh_is /usr/local/bin/tcsh
```

Input file (.dat)	**OPTIONS**	**Receiver depths (m)**	**# PLT points**
RSWT2a	N T J B	45	600
RSWT2b	N T J B	45	600
RSWT3a	P N T J B	45	600
RSWT3b	P N T J B	45	602
ASAwedgeE	K T C J	30, 150	1032
torusF	P N T J B	50, 170	549
torusE	P N T J B	50, 170	549
gilbert	N T J B	50	750
gilbertE	K S T J B	80	750
HorizForce	N V H T J P B 3	70	503
VertForce	N V H T J P B 2	35	503
PntSrc	N V H T J P B	35	503
TranspHF	N V H T J P B 2	35	471
TranspVF	N V H T J P B 2	35	471
mikeExD	P N T J B	100, 300	500
mikeExE	P N T J B	100, 300	1250
30degLBeam	P N T J B	100, 300	267
45degLBeam	P N T J B	100, 300	179
SLeeHC	P N T J B	35	2009
SLeet	P N V H T J B	35	1883
SLeed	P N T J B	35	1831
SLef	P N T J B	35	503
SLfe	P N T J B	35	503
norda3a25	P N T J B	50, 110	1250
norda3a25c2	P N T J F	50, 110	658
fnuw	P N V H T J B	30	654
SACL_circle_F	P N T J B	50, 170	191
SACL_step_F	P N T J B	50, 170	801

Table 2: Default parameter set up for the various test cases