

14. Apêndices

Plano de gestão de dados

Utilizamos o repositório de dados oficial da USP em: <https://repositorio.uspdigital.usp.br/> para as publicações referentes a este projeto. A intenção inicial era utilizar este local também para os códigos, porém o repositório da USP é somente para publicações, não havendo abertura para arquivos não publicados. Dessa forma, utilizamos o github (modo público) para os códigos, resultados e documentação detalhada (relatórios): <https://github.com/ssfuruie/cavitacao-proj-FAPESP-2021-04814-9/tree/main>.

Visando facilidades na compreensão dos códigos, usamos matlab e classes de objetos (orientados a objetos) para encapsulamento de problemas complexos. Em cada classe importante, há exemplos de utilização.

Visando facilidades na utilização dos programas, empregamos o conceito de tarefas que podem ser chamadas por um script simples, conforme documentado nas próximas seções. Em cada execução, gera-se um arquivo texto com os resultados obtidos.

Informamos também que todos os códigos e relatórios foram disponibilizados para os alunos e colaboradores do projeto.

Projeto e dados

O projeto proposto envolve a geração de muitos dados advindos de simulações, pois pretende-se inicialmente avaliar a abordagem por simulação. Serão simulações de sinais de cavitação e de propagação de ecos em meios heterogêneos. Para se avaliar a detecção, será necessário variar a amplitude relativa da cavitação, bem como o nível de ruído. Cada fase do desenvolvimento também implicará em vários testes para verificar a acurácia da implementação computacional.

Na fase experimental, para cada situação (distância do transmissor, tipo de microbolhas, fluxo das microbolhas, posição do receptor, tipo de transdutor, ...) serão medidas as pressões de excitação e as pressões incidentes nos transdutores em função do tempo. Estas tabelas serão arquivadas junto com os metadados.

Os resultados, relatórios, códigos e dados estão no github, no link acima.

Visão geral do programa mainSonoTrombolise.m

O programa está estruturado de modo a facilitar o seu uso por meio de diversas chamadas que podem ser realizadas em modo *batch*, especificando os valores dos parâmetros que definem o comportamento da simulação. O programa também pode ser rodado em modo interativo escolhendo a tarefa desejada, porém assumindo os valores default dos parâmetros, os quais podem ser modificados via alteração do código.

Um exemplo do modo batch com 2 chamadas ao programa:

```
% -----
argsin={'cav_simul_id','singlePointCombinedTypes',...           %cavitation type
'EchoAmpfactor',1,'StableAmpfactor',0.30,'InertAmpfactor',20.00,... %amplitudes
'NoiseLevel_dynRangeFactor',0.05,'FL',0.75,'FR',0.75,...       % noise level and band width
'opcao','i.2.0','result_suffix','inertialXstable_crit'};
mainSonoTrombolise(argsin);
% -----

% -----
argsin={'cav_simul_id','singlePointCombinedTypes',...           %cavitation type
'EchoAmpfactor',1,'StableAmpfactor',0.30,'InertAmpfactor',10.00,... %amplitudes
'NoiseLevel_dynRangeFactor',0.05,'FL',0.75,'FR',0.75,...       % noise level and band width
'opcao','i.2.0','result_suffix','inertialXstable_crit'};
mainSonoTrombolise(argsin);
% -----
```

E de modo interativo:

```
>> mainSonoTrombolise
Project SonoTrombolise - EPUSP/InCor, Version 3.12 (15/04/2022)
You have chosen as configuration option: byTask
a)Quick TEST(debugging) for rarefaction pressure and heat,2 foci(ok,tested;0m:56s):
[realistic grid; homogeneous medium; TRs uniformlyDistributed; foci:AtCenter_axial2; SensorIavg_region: ROI]
b)focus analysis at ROI center (central axial line contains a TR center)(ok,tested;0m:29s):
[realistic grid; homogeneous medium; TRs centeredTR; foci:AtCenter_axial; SensorIavg_region: ROI]
b.1)focus analysis at ROI center (central axial line does not necessarily contain a TR center)(ok,tested;0m:29s):
[realistic grid; homogeneous medium; TRs aroundCenter; foci:AtCenter_axial; SensorIavg_region: ROI]
b.2)same as b.1, but for non-homogeneous medium(ok,tested;6m:40s):
[realistic grid; boneLungPieces medium; TRs aroundCenter; foci:AtCenter_axial; SensorIavg_region: centralCuboide]
c)Temperature and ISPTA analysis in central cuboid region(ok,tested;8h:19m:00s):
[high res. grid; boneLungPieces medium; TRs centeredTR; foci:specific3; SensorIavg_region: centralCuboide]
c.1)Quick TEST, temperature and ISPTA analysis in central cuboid region(ok,tested;19m:52s):
[high res. grid; boneLungPieces medium; TRs centeredTR; foci:AtCenter_axial3; SensorIavg_region: centralCuboide]
d)Quick TEST for BONE detection,3x3 TRs,2 foci and quit(ok,tested;19m25s):
[high res. grid; boneLungPieces medium; TRs centeredTR; foci:AtCenter_axial2; SensorIavg_region: centralCuboide]
d.1)TEST for BONE detection,All TRs,2 foci and quit (ok,tested;2h25m:00s):
[high res. grid; boneLungPieces medium; TRs uniformlyDistributed; foci:AtCenter_axial2; SensorIavg_region: centralCuboide]
g)Typical echo spectrum for 1 TR. Narrow-band TR simulation via filter; Noise(set by TASK.NoiseLevel_dynRangeFactor). Focus at center,1TX,1RX and
quit(ok,tested;08m07s):
[high res. grid; boneLungPieces medium; TRs centeredTR; foci:AtCenter_axial; SensorIavg_region: centralCuboide]
h)Simulation of cavitation sources. Focus at center. Group of TR as receivers.(17m:00s):
[high res. grid; boneLungPieces medium; TRs centeredTR; foci:AtCenter_axial; Sensor_region: centralCuboide]
i)Analysis of cavitation(with cav.simulation). Group of TR as receivers.And quit():
[high res. grid; boneLungPieces medium; TRs centeredTR; foci:AtCenter_axial; Sensor_region: centralCuboide]
i.0.1)Test of ECHO sources. Echo signal and spectrum. Quit(< 1min):
[high res. grid; homogeneous medium; TRs centeredTR; foci:AtCenter_axial; Sensor_region: centralCuboide]
i.0.2)Test of STABLE cav sources. Cav signal and spectrum. Quit(< 1min):
[high res. grid; homogeneous medium; TRs centeredTR; foci:AtCenter_axial; Sensor_region: centralCuboide]
i.0.3)Test of INERTIAL cav sources. Cav signal and spectrum. Quit(< 1min):
[high res. grid; homogeneous medium; TRs centeredTR; foci:AtCenter_axial; Sensor_region: centralCuboide]
i.0.4)Test of combined sources. Source signal(amplitudes set by:E0,S0,I0) and spectrum. Quit(< 1min):
[high res. grid; homogeneous medium; TRs centeredTR; foci:AtCenter_axial; Sensor_region: centralCuboide]
i.1)Test of RECEPTION(source set by TASK.cav_simul_id).No TX source signal.No TR emulation.Noise(set by TASK.NoiseLevel_dynRangeFactor).Group
of TR as receivers. Quit(04m:00s):
[high res. grid; homogeneous medium; TRs centeredTR; foci:AtCenter_axial; Sensor_region: centralCuboide]
i.2)Test of RX EMULATION(source set by TASK.cav_simul_id).No TX source signal.Noise(set by TASK.NoiseLevel_dynRangeFactor).Group of TR as
receivers.And quit(04m:00s):
[high res. grid; homogeneous medium; TRs centeredTR; foci:AtCenter_axial; Sensor_region: centralCuboide]
i.2.0)Test of CAVITATION DETECTION;D&S in FREQ domain;(source set by TASK.cav_simul_id).No TX source signal.
Noise(set by TASK.NoiseLevel_dynRangeFactor).Group of TR as receivers.And quit(tested ok,< 3min)
[high res. grid; homogeneous medium; TRs centeredTR; foci:AtCenter_axial; Sensor_region: centralCuboide]
i.2.1)Test of CAVITATION DETECTION;D&S in TIME domain;(source set by TASK.cav_simul_id).No TX source signal.
Noise(set by TASK.NoiseLevel_dynRangeFactor).Group of TR as receivers.And quit(xx)
[high res. grid; homogeneous medium; TRs centeredTR; foci:AtCenter_axial; Sensor_region: centralCuboide]
i.3)Test: TX(echoes) and cavitation sources>window; TR emulation;Noise(set by TASK.NoiseLevel_dynRangeFactor);delay-and-sum;spectrum.Group of
TR as receivers.And quit(09m:00s):
[high res. grid; boneLungPieces medium; TRs centeredTR; foci:AtCenter_axial; Sensor_region: centralCuboide]
i.4)Test of cavitation sources and signals.No TX source signal.No TR emulation.Noise(set by TASK.NoiseLevel_dynRangeFactor).Group of TR as
receivers. Save signals, quit(4m:00s):
[high res. grid; homogeneous medium; TRs centeredTR; foci:AtCenter_axial; Sensor_region: centralCuboide]
p)rarefaction pressure and temperature analysis, homogeneous (ok,tested;07h:33m:00s)
[realistic grid; homogeneous medium; TRs uniformlyDistributed; foci:specific2; SensorIavg_region: ROI]
q)rarefaction pressure analysis,high res. grid, homogeneous,3 focus x-plane (ok,tested;02d01h48m00s)
[high res. grid; homogeneous medium; TRs uniformlyDistributed; foci:specific3; SensorIavg_region: ROI]
q.1)rarefaction pressure analysis, boneLungPieces,3 focus x-plane(ok,tested;08h:11m:00s)
[high res. grid; boneLungPieces medium; TRs uniformlyDistributed; foci:specific3; SensorIavg_region: centralCuboide]
q.2)rarefaction pressure analysis with BONE detection,3 focus x-plane(ok,tested;6h:10m:00s)
[high res. grid; boneLungPieces medium; TRs uniformlyDistributed; foci:specific3; SensorIavg_region: centralCuboide]
Obs.:Measurements such as p,p_rms,I_avg: require numMask.Nt values x (single or double)x(3 if I*avg), where numMask=number of nonzeros in
sensor.mask
-For quantities such as p_min_all, kwave returns values for all Nx,Ny,Nz voxels, regardless of sensor.mask
-If a task requires a transducer in the central axial region (e.g.:focus or intensity in central region), we enforce odd number of transducers in each
axis
-If a task uses high contrast medium, we have to use high resolution grid
-pre-defined settings so far (may be overridden by your choice):
TASK.cav_simul_id=singlePointCombinedTypes;
TASK.NoiseLevel_dynRangeFactor= 0.050 (factor for noise std)
```

-cited elapsed time is using LEB-SF2018 (Dell XPS8930,i7-8700CPU@3.20GHz,64GB,win10,C++,GPU Nvidia GTX1060,6GB)
Configurations(25) for a task:
Choose==>

O diagrama em blocos do sistema é a descrita na Figure 1.1 e aqui detalhada do ponto de vista de entrada de parâmetros (Figure 14.1), em especial para os blocos {D, E, F} para utilizar o programa.

O programa assumirá valores *default*, a menos que sejam passados via argumento na chamada do programa, por exemplo:

`mainSonoTrombolise('NoiseLevel_dynRangeFactor',0.05,'opcao','i.2.0');` neste caso estamos especificando o nível de ruído de 0.05 por meio da palavra-chave 'NoiseLevel_dynRangeFactor', e escolhendo a opção de tarefa 'i.2.0' por meio da palavra-chave 'opcao'. Os argumentos são sempre em dupla, especificando a palavra-chave e o seu valor. Caso não haja nenhum argumento, o programa rodará no modo interativo.

As palavras-chave estão definidas na function *getBatchInputPairValues.m*.

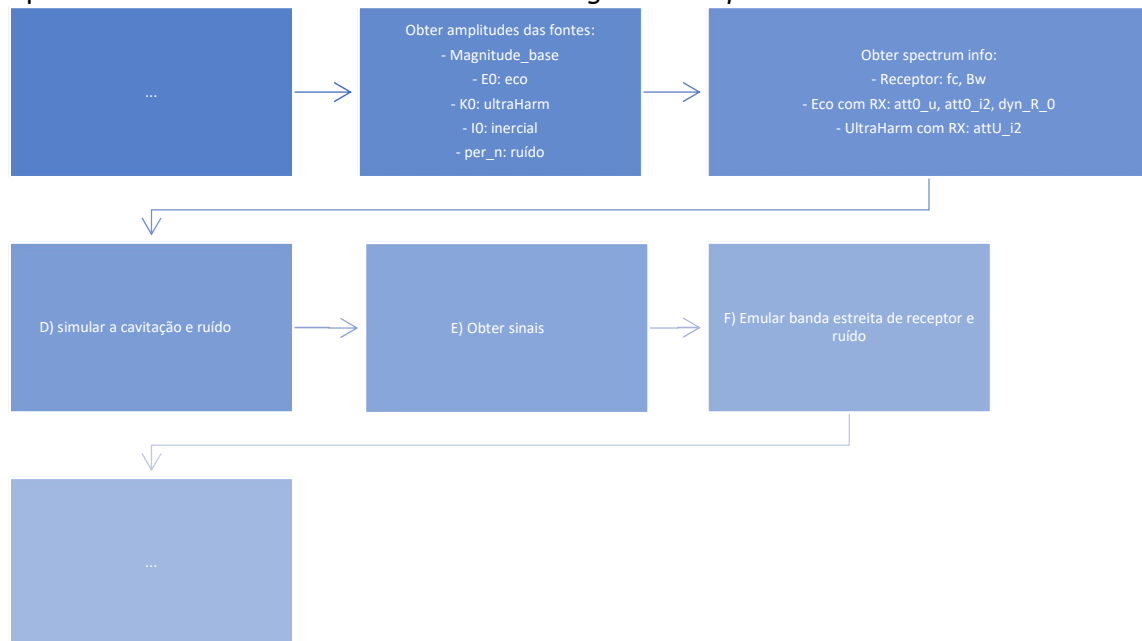


Figure 14.1 Visão global dos parâmetros de entrada para o programa.

Requisitos/Instruções para rodar o programa

INSTRUCTIONS to run this program (mainSonoTrombolise.m):

1. kwave should be installed, and kwave should be accessible by matlab (see Home=>Set Path)
2. set your computing environment (which directory to use for saving results).
Look for the word 'USER_is'; modify it; Create/modify the codes in 'switch(USER_is)' to set mainly some variables that depends on your computer such as DIR_PAI_DADOS. Otherwise, it will use defaults.
3. There are several parameters and constants that were pre-defined, such as 'SOURCE_MAGNITUDE','HEART_Xmin', ...
4. When you run, you'll see several pre-created tasks (look for 'getSonoTh_config_task')
5. The results (log report) are stored in the file 'results_temp.txt'. The program outputs lots of informations and figures that you can ignore. I just kept them because they are useful for debugging and reassurance.
-signals and calculated coherent signal are saved as .mat file if TASK.saveCoherentTemporalSignals=true;
(Obs.:I restricted to first focus because of amount of data. See line: if(TASK.SaveCoherentTemporalSignal==true ..)
-computed parameters for cavitation analysis are stored as .xlsx file. See line:xlsAppend_SF(xlsFileName,cell_1xN,header)
6. You can run this as a program (just type mainSonoTrombolise), or function/batch with arguments (pairs, see getBatchInputPairValues.m)