```
classdef FUNMAP_Input_exported < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                        matlab.ui.Figure
        Friction_____________________________Label_9  matlab.ui.control.Label
        Panel_32                        matlab.ui.container.Panel
        Button_41                       matlab.ui.control.Button
        Mpif90EditField                 matlab.ui.control.EditField
        Mpif90EditFieldLabel            matlab.ui.control.Label
        DoublePrecisionCheckBox         matlab.ui.control.CheckBox
        IntelCompilerCheckBox           matlab.ui.control.CheckBox
        ParallelModeCheckBox            matlab.ui.control.CheckBox
        MakefileEditField               matlab.ui.control.EditField
        MakefileEditFieldLabel          matlab.ui.control.Label
        Button_42                       matlab.ui.control.Button
        StartsimulationButton           matlab.ui.control.Button
        CreateMakefileButton            matlab.ui.control.Button
        TabGroup                        matlab.ui.container.TabGroup
        InputTab                        matlab.ui.container.Tab
        Friction_____________________________Label_10  matlab.ui.control.Label
        Panel_33                        matlab.ui.container.Panel
        SteadyTimeEditField             matlab.ui.control.NumericEditField
        SteadyTimeEditFieldLabel        matlab.ui.control.Label
        TimeintervalsecEditField        matlab.ui.control.NumericEditField
        TimeintervalsecEditFieldLabel   matlab.ui.control.Label
        UseSmagorinskyCheckBox          matlab.ui.control.CheckBox
        GenerateinputtxtButton          matlab.ui.control.Button
        Friction_____________________________Label_7  matlab.ui.control.Label
        Panel_30                        matlab.ui.container.Panel
        C2EditField                     matlab.ui.control.NumericEditField
        C2EditFieldLabel                matlab.ui.control.Label
        C1EditFieldLabel                matlab.ui.control.Label
        C1EditField                     matlab.ui.control.NumericEditField
        ButtonGroup_23                  matlab.ui.container.ButtonGroup
        ViscositybreakingButton         matlab.ui.control.RadioButton
        ShockwavecapturingButton        matlab.ui.control.RadioButton
        DimensionLabel                  matlab.ui.control.Label
        DepthLabel                      matlab.ui.control.Label
        Friction_____________________________Label_6  matlab.ui.control.Label
        Panel_29                        matlab.ui.container.Panel
        SavefilestoEditField            matlab.ui.control.EditField
        SavefilestoLabel                matlab.ui.control.Label
        Button_14                       matlab.ui.control.Button
        Friction_____________________________Label_5  matlab.ui.control.Label
        Panel_28                        matlab.ui.container.Panel
        SpacingEditField                matlab.ui.control.NumericEditField
        SpacingEditFieldLabel           matlab.ui.control.Label
        BoundarylimitsButton            matlab.ui.control.Button
        WestEditField_2                 matlab.ui.control.NumericEditField
        SouthEditField_2                matlab.ui.control.NumericEditField
        EastEditField_2                 matlab.ui.control.NumericEditField
        NorthEditField_2                matlab.ui.control.NumericEditField
```

```
54          ImportgaugelistCheckBox          matlab.ui.control.CheckBox
55          LoadFileEditField                matlab.ui.control.EditField
56          Button_7                         matlab.ui.control.Button
57          CreatenestedgridboundaryCheckBox  matlab.ui.control.CheckBox
58          BottomFrictionLabel              matlab.ui.control.Label
59          Panel_26                         matlab.ui.container.Panel
60          ConstantFrictionValue            matlab.ui.control.NumericEditField
61          ValueEditFieldLabel              matlab.ui.control.Label
62          Button_10                        matlab.ui.control.Button
63          FrictionFileEditField            matlab.ui.control.EditField
64          FrictionFileEditFieldLabel       matlab.ui.control.Label
65          ButtonGroup_8                    matlab.ui.container.ButtonGroup
66          NonconstantButton                matlab.ui.control.RadioButton
67          ConstantCoefficientButton        matlab.ui.control.RadioButton
68          PhysicsLabel                     matlab.ui.control.Label
69          ObstaclesLabel                   matlab.ui.control.Label
70          Panel_24                         matlab.ui.container.Panel
71          AddobstaclesCheckBox             matlab.ui.control.CheckBox
72          ObstacleFilename                 matlab.ui.control.EditField
73          ObstacleLoadFile                 matlab.ui.control.Button
74          SpongeBoundaryLabel              matlab.ui.control.Label
75          Panel_23                         matlab.ui.container.Panel
76          MaximumCspEditField              matlab.ui.control.NumericEditField
77          MaximumCspEditFieldLabel         matlab.ui.control.Label
78          MaximumCdEditField               matlab.ui.control.NumericEditField
79          MaximumCdEditFieldLabel          matlab.ui.control.Label
80          DecayTypeDropDown                matlab.ui.control.DropDown
81          DecayTypeDropDownLabel           matlab.ui.control.Label
82          SouthLabel                       matlab.ui.control.Label
83          WestLabel                        matlab.ui.control.Label
84          EastLabel                        matlab.ui.control.Label
85          NorthLabel                       matlab.ui.control.Label
86          WestEditField                    matlab.ui.control.NumericEditField
87          SouthEditField                   matlab.ui.control.NumericEditField
88          EastEditField                    matlab.ui.control.NumericEditField
89          NorthEditField                   matlab.ui.control.NumericEditField
90          DiffusionCheckBox                matlab.ui.control.CheckBox
91          FrictionCheckBox                 matlab.ui.control.CheckBox
92          DirectCheckBox                   matlab.ui.control.CheckBox
93          LayerwidthLabel                  matlab.ui.control.Label
94          GridLabel                        matlab.ui.control.Label
95          Panel_22                         matlab.ui.container.Panel
96          GridSizeLabel                    matlab.ui.control.Label
97          SouthwestcornerLabel             matlab.ui.control.Label
98          ButtonGroup_4                    matlab.ui.container.ButtonGroup
99          Button_2                         matlab.ui.control.RadioButton
100         CartesianButton                  matlab.ui.control.RadioButton
101         SphericalButton                  matlab.ui.control.RadioButton
102         LatitudeEditField                matlab.ui.control.NumericEditField
103         LatitudeEditFieldLabel           matlab.ui.control.Label
104         LongitudeEditField               matlab.ui.control.NumericEditField
105         LongitudeEditFieldLabel          matlab.ui.control.Label
106         YEditField                       matlab.ui.control.NumericEditField
107         YgridsizeLabel                   matlab.ui.control.Label
108         XEditField                       matlab.ui.control.NumericEditField
109         XEditFieldLabel                  matlab.ui.control.Label
110         TimeLabel_2                      matlab.ui.control.Label
111         Panel_21                         matlab.ui.container.Panel
```

```
112         TotalEditField                  matlab.ui.control.NumericEditField
113         TotalEditFieldLabel             matlab.ui.control.Label
114         SimulationIntervalEditField     matlab.ui.control.NumericEditField
115         SimulationIntervalEditFieldLabel  matlab.ui.control.Label
116         GaugeIntervalEditField          matlab.ui.control.NumericEditField
117         GaugeIntervalEditFieldLabel     matlab.ui.control.Label
118         StartEditField                  matlab.ui.control.NumericEditField
119         ScreenLoadingIntervalLabel      matlab.ui.control.Label
120         PrintLabel                      matlab.ui.control.Label
121         Panel_20                        matlab.ui.container.Panel
122         ResultsfolderEditField          matlab.ui.control.EditField
123         ResultsfolderEditFieldLabel     matlab.ui.control.Label
124         Panel_19                        matlab.ui.container.Panel
125         XEditField_2                    matlab.ui.control.NumericEditField
126         XEditField_2Label               matlab.ui.control.Label
127         YEditField_2                    matlab.ui.control.NumericEditField
128         YEditField_2Label               matlab.ui.control.Label
129         Panel_18                        matlab.ui.container.Panel
130         AngleEditField                  matlab.ui.control.NumericEditField
131         AngleEditFieldLabel             matlab.ui.control.Label
132         StartatXEditField               matlab.ui.control.NumericEditField
133         StartatXEditFieldLabel          matlab.ui.control.Label
134         DepthmEditFieldLabel            matlab.ui.control.Label
135         DepthmEditField                 matlab.ui.control.NumericEditField
136         InvertvaluesCheckBox            matlab.ui.control.CheckBox
137         WaterlevelmEditFieldLabel       matlab.ui.control.Label
138         WaterlevelmEditField            matlab.ui.control.NumericEditField
139         Button_8                        matlab.ui.control.Button
140         DepthFileEditField              matlab.ui.control.EditField
141         ButtonGroup_7                   matlab.ui.container.ButtonGroup
142         Button4                         matlab.ui.control.RadioButton
143         SlopeButton                     matlab.ui.control.RadioButton
144         FlatButton                      matlab.ui.control.RadioButton
145         ImportDataButton                matlab.ui.control.RadioButton
146         Friction_____________________________Label_2  matlab.ui.control.↙
Label
147         ParallelInfoLabel               matlab.ui.control.Label
148         Panel_17                        matlab.ui.container.Panel
149         Panel_5                         matlab.ui.container.Panel
150         ProcessorY                      matlab.ui.control.NumericEditField
151         YLabel                          matlab.ui.control.Label
152         ProcessorX                      matlab.ui.control.NumericEditField
153         ProcessornumbersXLabel          matlab.ui.control.Label
154         Panel_25                        matlab.ui.container.Panel
155         ButtonGroup_3                   matlab.ui.container.ButtonGroup
156         FroudeCapEditField              matlab.ui.control.NumericEditField
157         FroudeCapEditFieldLabel         matlab.ui.control.Label
158         NonlinearShallowWaterEquationButton  matlab.ui.control.RadioButton
159         Button                          matlab.ui.control.RadioButton
160         FullyNonlinearBoussinesqEquationButton  matlab.ui.control.RadioButton
161         LinearShallowWaterEquationButton  matlab.ui.control.RadioButton
162         Panel_27                        matlab.ui.container.Panel
163         BottomFrictionEditField         matlab.ui.control.NumericEditField
164         BottomFrictionEditFieldLabel    matlab.ui.control.Label
165         WetDrySchemeEditField           matlab.ui.control.NumericEditField
166         WetDrySchemeEditFieldLabel      matlab.ui.control.Label
167         InitialConditionTab             matlab.ui.container.Tab
168         SedimentTransportCheckBox       matlab.ui.control.CheckBox
```

```
169         GenerateinputtxtButton_4        matlab.ui.control.Button
170         ButtonGroup_18                  matlab.ui.container.ButtonGroup
171         VesselButton                    matlab.ui.control.RadioButton
172         MeteotsunamiButton              matlab.ui.control.RadioButton
173         Button_31                       matlab.ui.control.RadioButton
174         WavemakerButton                 matlab.ui.control.RadioButton
175         ImportUVZButton                 matlab.ui.control.RadioButton
176         SelectInitialConditonLabel      matlab.ui.control.Label
177         TabGroup2                       matlab.ui.container.TabGroup
178         Tab_5                           matlab.ui.container.Tab
179         Tab                             matlab.ui.container.Tab
180         Button_33                       matlab.ui.control.Button
181         VVelocityEditField              matlab.ui.control.EditField
182         VVelocityEditFieldLabel         matlab.ui.control.Label
183         UVelocityEditField              matlab.ui.control.EditField
184         UVelocityEditFieldLabel         matlab.ui.control.Label
185         SurfaceHeightEditField          matlab.ui.control.EditField
186         SurfaceHeightEditFieldLabel     matlab.ui.control.Label
187         Button_34                       matlab.ui.control.Button
188         Button_32                       matlab.ui.control.Button
189         ImportUVZfilesLabel             matlab.ui.control.Label
190         Label_9                         matlab.ui.control.Label
191         Tab_2                           matlab.ui.container.Tab
192         WaveHeightmEditField            matlab.ui.control.NumericEditField
193         WaveHeightmEditFieldLabel       matlab.ui.control.Label
194         MaximumEditField                matlab.ui.control.NumericEditField
195         MaximumEditFieldLabel           matlab.ui.control.Label
196         MinimumEditField                matlab.ui.control.NumericEditField
197         MinimumEditFieldLabel           matlab.ui.control.Label
198         PeakEditField                   matlab.ui.control.NumericEditField
199         FrequencyPeakLabel              matlab.ui.control.Label
200         FrequencyLabel                  matlab.ui.control.Label
201         WaveCompCountEditField          matlab.ui.control.NumericEditField
202         WavecomponentcountEditFieldLabel  matlab.ui.control.Label
203         WavecomponentfileEditField      matlab.ui.control.EditField
204         WavecomponentfileEditFieldLabel  matlab.ui.control.Label
205         ThetadegreesEditField           matlab.ui.control.NumericEditField
206         ThetadegreesEditFieldLabel      matlab.ui.control.Label
207         PeriodsecEditField              matlab.ui.control.NumericEditField
208         PeriodsecEditFieldLabel         matlab.ui.control.Label
209         WidthDeltaEditField             matlab.ui.control.NumericEditField
210         WidthDeltaEditFieldLabel        matlab.ui.control.Label
211         TimerampsecEditField            matlab.ui.control.NumericEditField
212         TimerampsecEditFieldLabel       matlab.ui.control.Label
213         WidthmEditField                 matlab.ui.control.NumericEditField
214         WidthmEditFieldLabel            matlab.ui.control.Label
215         YcoordinatemEditField           matlab.ui.control.NumericEditField
216         YcoordinatemEditFieldLabel      matlab.ui.control.Label
217         XcoordinatemEditField           matlab.ui.control.NumericEditField
218         XcoordinatemEditFieldLabel      matlab.ui.control.Label
219         LagtimesecEditField             matlab.ui.control.NumericEditField
220         LagtimesecEditFieldLabel        matlab.ui.control.Label
221         WaterdepthmEditField            matlab.ui.control.NumericEditField
222         WaterdepthmEditFieldLabel       matlab.ui.control.Label
223         AmplitudemEditField             matlab.ui.control.NumericEditField
224         AmplitudemEditFieldLabel        matlab.ui.control.Label
225         WavemakertypeDropDown           matlab.ui.control.DropDown
226         WavemakertypeDropDownLabel      matlab.ui.control.Label
```

```
227         Button_20                          matlab.ui.control.Button
228         WavemakerParametersLabel_2         matlab.ui.control.Label
229         Label_10                           matlab.ui.control.Label
230         Tab_4                              matlab.ui.container.Tab
231         WindLabel                          matlab.ui.control.Label
232         PressureLabel                      matlab.ui.control.Label
233         Button_43                          matlab.ui.control.Button
234         PressurefileEditField              matlab.ui.control.EditField
235         PressurefileEditFieldLabel         matlab.ui.control.Label
236         StormfileEditFieldLabel            matlab.ui.control.Label
237         StormfileEditField                 matlab.ui.control.EditField
238         WindwaveinteractionSwitch          matlab.ui.control.Switch
239         WindwaveinteractionSwitchLabel     matlab.ui.control.Label
240         HollandstormmodelSwitch            matlab.ui.control.Switch
241         HollandstormmodelSwitchLabel       matlab.ui.control.Label
242         ConstantwindfileEditField          matlab.ui.control.EditField
243         ConstantwindfileEditFieldLabel     matlab.ui.control.Label
244         ConstantwindfieldSwitch            matlab.ui.control.Switch
245         ConstantwindfieldSwitchLabel       matlab.ui.control.Label
246         CrestRatioEditField                matlab.ui.control.NumericEditField
247         CrestRatioEditFieldLabel           matlab.ui.control.Label
248         WindstresscoefficientEditFieldLabel  matlab.ui.control.Label
249         WindstresscoefficientEditField     matlab.ui.control.NumericEditField
250         AirpressureSwitch                  matlab.ui.control.Switch
251         AirpressureSwitchLabel             matlab.ui.control.Label
252         WindforceSwitch                    matlab.ui.control.Switch
253         WindforceSwitchLabel               matlab.ui.control.Label
254         Button_30                          matlab.ui.control.Button
255         Button_29                          matlab.ui.control.Button
256         MeteotsunamiLabel                  matlab.ui.control.Label
257         Label_11                           matlab.ui.control.Label
258         Tab_3                              matlab.ui.container.Tab
259         AddrollereffectsCheckBox           matlab.ui.control.CheckBox
260         TimeStepSedToMorphlEditField       matlab.ui.control.NumericEditField
261         TimeStepSedToMorpEditFieldLabel  matlab.ui.control.Label
262         ButtonGroup_21                     matlab.ui.container.ButtonGroup
263         Button_37                          matlab.ui.control.RadioButton
264         SedimentbedButton                  matlab.ui.control.RadioButton
265         HardbednoerosionButton             matlab.ui.control.RadioButton
266         BottomTypeLabel                    matlab.ui.control.Label
267         Button_35                          matlab.ui.control.Button
268         FileEditField                      matlab.ui.control.EditField
269         FileEditFieldLabel                 matlab.ui.control.Label
270         RungeKuttaparam2EditField_2        matlab.ui.control.NumericEditField
271         RungeKuttaparam2EditFieldLabel_2  matlab.ui.control.Label
272         ButtonGroup_22                     matlab.ui.container.ButtonGroup
273         Button_38                          matlab.ui.control.RadioButton
274         NoButton_7                         matlab.ui.control.RadioButton
275         YesButton_8                        matlab.ui.control.RadioButton
276         ConsideravalancheLabel             matlab.ui.control.Label
277         MorphologicalChangeLabel_2         matlab.ui.control.Label
278         AvalancheLabel                     matlab.ui.control.Label
279         TimeStepBouss                      matlab.ui.control.NumericEditField
280         EditFieldLabel                     matlab.ui.control.Label
281         BedChangeLabel                     matlab.ui.control.Label
282         ButtonGroup_20                     matlab.ui.container.ButtonGroup
283         Button_36                          matlab.ui.control.RadioButton
284         NobedchangeButton                  matlab.ui.control.RadioButton
```

```
        UpdatedepthButton                matlab.ui.control.RadioButton
        MinDepthSedPickUpEditField       matlab.ui.control.NumericEditField
        MinDepthSedPickUpEditFieldLabel  matlab.ui.control.Label
        RungeKuttaparameter2EditField    matlab.ui.control.NumericEditField
        RungeKuttaparameter2EditFieldLabel  matlab.ui.control.Label
        RungeKuttaparameter1EditField    matlab.ui.control.NumericEditField
        RungeKuttaparameter1EditFieldLabel  matlab.ui.control.Label
        ShieldsparamBedloadEditField     matlab.ui.control.NumericEditField
        ShieldsparameterbedloadEditFieldLabel  matlab.ui.control.Label
        CriticalShieldsEditField         matlab.ui.control.NumericEditField
        CriticalShieldsEditFieldLabel    matlab.ui.control.Label
        SettlingvelocitymsEditField      matlab.ui.control.NumericEditField
        SettlingvelocitymsEditFieldLabel  matlab.ui.control.Label
        SedimentporosityEditField        matlab.ui.control.NumericEditField
        SedimentporosityEditFieldLabel   matlab.ui.control.Label
        SedimentdensityEditField         matlab.ui.control.NumericEditField
        SedimentdensityEditFieldLabel    matlab.ui.control.Label
        MediangrainDMEditField           matlab.ui.control.NumericEditField
        MediangrainDMEditFieldLabel      matlab.ui.control.Label
        NumericalschemeLabel             matlab.ui.control.Label
        ButtonGroup_19                   matlab.ui.container.ButtonGroup
        TVDButton                        matlab.ui.control.RadioButton
        UpwindingButton                  matlab.ui.control.RadioButton
        SedimentLabel                    matlab.ui.control.Label
        Label_12                         matlab.ui.control.Label
        Label_5                          matlab.ui.control.Label
        Label_6                          matlab.ui.control.Label
        Tab_6                            matlab.ui.container.Tab
        SelectMethodLabel                matlab.ui.control.Label
        ValueEditField                   matlab.ui.control.NumericEditField
        ValueEditFieldLabel_2            matlab.ui.control.Label
        FrictionCoefficientEditField     matlab.ui.control.NumericEditField
        FrictionCoefficientEditFieldLabel  matlab.ui.control.Label
        ViscosityCheckBox                matlab.ui.control.CheckBox
        FrictionCheckBox_2               matlab.ui.control.CheckBox
        ShockcapturingCheckBox           matlab.ui.control.CheckBox
        ActivateCheckBox                 matlab.ui.control.CheckBox
        IncludeSedimentEffectintheModelCheckBox  matlab.ui.control.CheckBox
        MinimumClearanceEditField        matlab.ui.control.NumericEditField
        MinimumClearanceLabel            matlab.ui.control.Label
        DeepDraftVesselsLabel            matlab.ui.control.Label
        Label_8                          matlab.ui.control.Label
        VesselLabel                      matlab.ui.control.Label
        Button_40                        matlab.ui.control.Button
        ImportvesselfilesTextArea        matlab.ui.control.TextArea
        ImportVesselFilesLabel           matlab.ui.control.Label
        Label_13                         matlab.ui.control.Label
        ExpectedOutputTab                matlab.ui.container.Tab
        GridLayout                       matlab.ui.container.GridLayout
        ThresholdEditField               matlab.ui.control.NumericEditField
        ThresholdEditFieldLabel          matlab.ui.control.Label
        MaximumWaveHeightCheckBox        matlab.ui.control.CheckBox
        MinimumWaveHeightCheckBox        matlab.ui.control.CheckBox
        VVelocitymeanCheckBox            matlab.ui.control.CheckBox
        UVelocitymeanCheckBox            matlab.ui.control.CheckBox
        GenerateinputtxtButton_2         matlab.ui.control.Button
        OthersLabel                      matlab.ui.control.Label
        TimeLabel                        matlab.ui.control.Label
```

```
        FluxLabel                       matlab.ui.control.Label
        MaskLabel                       matlab.ui.control.Label
        SourceLabel                     matlab.ui.control.Label
        WaveHeightLabel                 matlab.ui.control.Label
        VelocityLabel                   matlab.ui.control.Label
        UndertowCheckBox                matlab.ui.control.CheckBox
        MaximumVelocityCheckBox         matlab.ui.control.CheckBox
        BreakingAgeCheckBox             matlab.ui.control.CheckBox
        GyFluxCheckBox                  matlab.ui.control.CheckBox
        GxFluxCheckBox                  matlab.ui.control.CheckBox
        FyFluxCheckBox                  matlab.ui.control.CheckBox
        FxFluxCheckBox                  matlab.ui.control.CheckBox
        QmomentumFluxCheckBox           matlab.ui.control.CheckBox
        PMomentumFluxCheckBox           matlab.ui.control.CheckBox
        BreakingLocationCheckBox        matlab.ui.control.CheckBox
        YSourceCheckBox                 matlab.ui.control.CheckBox
        PressureFieldCheckBox           matlab.ui.control.CheckBox
        XSourceCheckBox                 matlab.ui.control.CheckBox
        RollerinducedFluxCheckBox       matlab.ui.control.CheckBox
        WetdrymaskforBoussinesqNSWECheckBox  matlab.ui.control.CheckBox
        WaveHeightCheckBox              matlab.ui.control.CheckBox
        WetdrymaskCheckBox              matlab.ui.control.CheckBox
        SurfaceElevationattimetCheckBox  matlab.ui.control.CheckBox
        ArrivalCheckBox                 matlab.ui.control.CheckBox
        VVelocityCheckBox               matlab.ui.control.CheckBox
        MaxMomentumFluxCheckBox         matlab.ui.control.CheckBox
        UVelocityCheckBox               matlab.ui.control.CheckBox
        MaxVorticityCheckBox            matlab.ui.control.CheckBox
        DepthCheckBox                   matlab.ui.control.CheckBox
        TabGroup3                       matlab.ui.container.TabGroup
        LogReportTab                    matlab.ui.container.Tab
        Preview_LogReport               matlab.ui.control.TextArea
        inputtxtTab                     matlab.ui.container.Tab
        Preview_Input                   matlab.ui.control.TextArea
        Makefile                        matlab.ui.container.Tab
        Preview_Makefile                matlab.ui.control.TextArea
    end


    properties (Access = private)
        filename
        path
        inputpath
        inputdefault
        outputfile
        ouputdirectoryfinal
        SWE
        DspongeSwitch
        DspongeDecay
        initialHeight
        initialXvelocity
        initialYvelocity
        Mglob
        Nglob
        NumberStations
        STATION_FILE
        bathymetryinputdata
        NestStation
```

```
401         WORKFOLDER
402         GAUGEFILE
403         DEPTHFILE
404         GaugeListGrids
405         GaugeFname
406         Cd_file
407         filenamefriction
408         SurfaceHeightNAME
409         XVelocityNAME
410         YVelocityNAME
411         SurfaceHeightLOC
412         XVelocityLOC
413         YVelocityLOC
414         obstacleLOC
415         obstacleNAME
416         WaveFileNAME
417         WaveFileLOC
418         wavemaker
419         ConsWindNAME
420         ConsWindLOC
421         StormFileNAME
422         StormFileLOC
423         PressureFileNAME
424         PressureFileLOC
425         HardBottomFileNAME
426         HardBottomFileLOC
427         VesselNumber
428         VesselLOC
429         VesselNAME
430         HardbottomLOC
431         HardbottomNAME
432         CouplingLOC
433         CouplingNAME
434         mpif90val
435         directory
436         logreport
437         MakefileContent
438         processPID
439         makefile
440         depthfilelocation
441         depthdata
442         depthtype
443         rasterdegrees
444         depthfinal
445         GAUGEFILEllocation
446         colsChildLayer
447         rowsChildLayer
448         directorylog
449         CouplingLOC2
450         fortranFilePath0
451         gfortranReport
452         resultCompile
453         resultRun
454         inputFolder
455         CouplingX
456         CouplingY
457         CoupX
458         CoupY
```

```
459         GAUGEFILElocation
460         latGauge
461         gaugetext0
462         CoupMatrixON
463         gridX
464         gridY
465         row1
466         col1
467     end
468 
469 
470     % Callbacks that handle component events
471     methods (Access = private)
472 
473         % Code that executes after component creation
474         function startupFcn(app)
475                 %Set the screenshot mode to 'manual' to disable automatic
screenshots,
476      %to improve app efficiency
477             currentAppModel.MetadataModel.ScreenshotMode = 'manual';
478         end
479 
480         % Button pushed function: Button_14
481         function Button_14Pushed(app, event)
482             workingfolder = uigetdir;
483             workingfolder = fullfile(workingfolder);
484 
485             %Create the 'INPUT_FILES' folder in the working directory
486             app.WORKFOLDER = fullfile(string(workingfolder), 'INPUT_FILES');
487             app.SavefilestoEditField.Value = fullfile(string(workingfolder),
'INPUT_FILES');
488             app.SavefilestoEditField.FontColor = 'k';
489         end
490 
491         % Button pushed function: Button_8
492         function Button_8Pushed(app, event)
493             %Load the bathymetry file and display its filename in the textbox
494             [bathymetryfile,path2] = uigetfile('*.txt;*.tif');
495             app.depthfilelocation  = string(path2);
496             app.bathymetryinputdata = string(path2) + string(bathymetryfile);
497             app.DepthFileEditField.BackgroundColor = 'w';
498             app.DepthFileEditField.FontColor =  'k';
499 
500 %             try
501               if contains(string(app.bathymetryinputdata),'.txt')
502                   app.depthtype = 1;
503                   A1 = readmatrix(app.bathymetryinputdata);
504                   app.depthdata = A1;
505                   app.DEPTHFILE = string(bathymetryfile);
506 
507                   %Extract M and N global dimensions
508                   [row,col] = size(A1);
509                   app.XEditField_2.Value = col;
510                   app.YEditField_2.Value = row;
511                   app.XEditField_2.FontColor = 'k';
512                   app.YEditField_2.FontColor = 'k';
513                   app.ButtonGroup_4.SelectedObject = app.Button_2;
514 
```

```
515                 elseif contains(string(app.bathymetryinputdata),'.tif')
516                     %Extract information from the bathymetry
517                     app.depthtype = 2;
518                     [A1,R2] = readgeoraster(app.bathymetryinputdata);
519                     A =  size(A1);
520                     app.depthdata = flipud(A1); %FUNWAVE reads the depth file↙
starting from the southwest corner
521                     %The readgeoraster function generates a matrix with the↙
first point representing the northwest corner
522 
523                     bathname= erase(bathymetryfile, '.tif');  % Remove .tif↙
extension
524                     app.DEPTHFILE = [bathname, '.txt'];       % Append .txt↙
extension
525 
526 
527                     %Extract lat and long of the southwest corner
528                     app.LongitudeEditField.Value = min(R2.LongitudeLimits);
529                     app.LatitudeEditField.Value = min(R2.LatitudeLimits);
530                     app.LongitudeEditField.FontColor =  'k';
531                     app.LatitudeEditField.FontColor = 'k';
532                     app.LatitudeEditField.Enable = "on";
533                     app.LongitudeEditField.Enable = "on";
534 
535                     %Extract grid size
536                     app.XEditField.Value = R2.CellExtentInLatitude;
537                     app.YEditField.Value = R2.CellExtentInLongitude;
538                     app.gridX =  R2.CellExtentInLatitude;
539                     app.gridY = R2.CellExtentInLongitude;
540                     app.XEditField.FontColor = 'k';
541                     app.YEditField.FontColor = 'k';
542 
543                     %Extract M and N global dimenstions
544                     app.XEditField_2.Value = A(1,2);
545                     app.YEditField_2.Value = A(1,1);
546                     app.col1 = A(1,2);
547                     app.row1 = A(1,1);
548                     app.XEditField_2.FontColor = 'k';
549                     app.YEditField_2.FontColor = 'k';
550 
551                     if R2.AngleUnits == "degrees"
552                         app.ButtonGroup_4.SelectedObject = app.↙
SphericalButton;
553                         app.SphericalButton.FontColor = 'k';
554                         app.CartesianButton.FontColor = 'k';
555                     end
556                     app.rasterdegrees = R2;
557                 else
558                     app.ButtonGroup_4.SelectedObject = app.Button_2;
559                     app.DepthFileEditField.Value = '';
560                 end
561                     app.DepthFileEditField.Value = app.DEPTHFILE; %Display the↙
filename in the textbox
562 %             end
563 
564         end
565 
566         % Callback function
```

```
 567         function TotalTimeEditFieldValueChanged(app, event)
 568
 569         end
 570
 571         % Callback function
 572         function PlotintervalEditFieldValueChanged(app, event)
 573
 574
 575         end
 576
 577         % Button pushed function: GenerateinputtxtButton
 578         function GenerateinputtxtButtonPushed(app, event)
 579 %Please cite this paper when you use the applications:
 580 %Felix, R., Tan, E. H. Z., Watanabe, M., Verolino, A., Puah, J. Y., & Switzer, A. D. (2025). Funwave-based open-access mapping applications (FUNMAP) applied to Tsunami modelling from the Manila Trench to Manila Bay, Philippines. Geoscience Letters. https://doi.org/10.1186/s40562-025-00422-5
 581
 582
 583
 584
 585 % NANYANG TECHNOLOGICAL UNIVERSITY - NTUITIVE PTE LTD (NTUITIVE) Dual License Agreement
 586 % Non-Commercial Use Only
 587 % This NTUITIVE License Agreement, including all exhibits ("NTUITIVE-LA") is a legal agreement between you and NTUITIVE (or "we") located at 71 Nanyang Drive, NTU Innovation Centre, #01-109, Singapore 637722, a wholly owned subsidiary of Nanyang Technological University ("NTU") for the software or data identified above, which may include source code, and any associated materials, text or speech files, associated media and "online" or electronic documentation and any updates we provide in our discretion (together, the "Software").
 588 %
 589 % By installing, copying, or otherwise using this Software, found at https://github.com/raquelpfelix, you agree to be bound by the terms of this NTUITIVE-LA. If you do not agree, do not install copy or use the Software. The Software is protected by copyright and other intellectual property laws and is licensed, not sold.   If you wish to obtain a commercial royalty bearing license to this software please contact us at rf761@bath.ac.uk.
 590 %
 591 % SCOPE OF RIGHTS:
 592 % You may use, copy, reproduce, and distribute this Software for any non-commercial purpose, subject to the restrictions in this NTUITIVE-LA. Some purposes which can be non-commercial are teaching, academic research, public demonstrations and personal experimentation. You may also distribute this Software with books or other teaching materials, or publish the Software on websites, that are intended to teach the use of the Software for academic or other non-commercial purposes.
 593 % You may not use or distribute this Software or any derivative works in any form for commercial purposes. Examples of commercial purposes would be running business operations, licensing, leasing, or selling the Software, distributing the Software for use with commercial products, using the Software in the creation or use of commercial products or any other activity which purpose is to procure a commercial gain to you or others.
 594 % If the Software includes source code or data, you may create derivative works of such portions of the Software and distribute the modified Software for non-commercial purposes, as provided herein.
 595 % If you distribute the Software or any derivative works of the Software, you will distribute them under the same terms and conditions as in this license, and you will not grant other rights to the Software or derivative works that are
```

different from those provided by this NTUITIVE-LA.
% If you have created derivative works of the Software, and distribute such derivative works, you will cause the modified files to carry prominent notices so that recipients know that they are not receiving the original Software. Such notices must state: (i) that you have changed the Software; and (ii) the date of any changes.
%
% You may not distribute this Software or any derivative works.
% In return, we simply require that you agree:
% 1. That you will not remove any copyright or other notices from the Software.
% 2. That if any of the Software is in binary format, you will not attempt to modify such portions of the Software, or to reverse engineer or decompile them, except and only to the extent authorized by applicable law.
% 3. That NTUITIVE is granted back, without any restrictions or limitations, a non-exclusive, perpetual, irrevocable, royalty-free, assignable and sub-licensable license, to reproduce, publicly perform or display, install, use, modify, post, distribute, make and have made, sell and transfer your modifications to and/or derivative works of the Software source code or data, for any purpose.
% 4. That any feedback about the Software provided by you to us is voluntarily given, and NTUITIVE shall be free to use the feedback as it sees fit without obligation or restriction of any kind, even if the feedback is designated by you as confidential.
% 5. THAT THE SOFTWARE COMES "AS IS", WITH NO WARRANTIES. THIS MEANS NO EXPRESS, IMPLIED OR STATUTORY WARRANTY, INCLUDING WITHOUT LIMITATION, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, ANY WARRANTY AGAINST INTERFERENCE WITH YOUR ENJOYMENT OF THE SOFTWARE OR ANY WARRANTY OF TITLE OR NON-INFRINGEMENT. THERE IS NO WARRANTY THAT THIS SOFTWARE WILL FULFILL ANY OF YOUR PARTICULAR PURPOSES OR NEEDS. ALSO, YOU MUST PASS THIS DISCLAIMER ON WHENEVER YOU DISTRIBUTE THE SOFTWARE OR DERIVATIVE WORKS.
% 6. THAT NEITHER NTUITIVE NOR NTU NOR ANY CONTRIBUTOR TO THE SOFTWARE WILL BE LIABLE FOR ANY DAMAGES RELATED TO THE SOFTWARE OR THIS NTUITIVE-LA, INCLUDING DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL OR INCIDENTAL DAMAGES, TO THE MAXIMUM EXTENT THE LAW PERMITS, NO MATTER WHAT LEGAL THEORY IT IS BASED ON. ALSO, YOU MUST PASS THIS LIMITATION OF LIABILITY ON WHENEVER YOU DISTRIBUTE THE SOFTWARE OR DERIVATIVE WORKS.
% 7. That we have no duty of reasonable care or lack of negligence, and we are not obligated to (and will not) provide technical support for the Software.
% 8. That if you breach this NTUITIVE-LA or if you sue anyone over patents that you think may apply to or read on the Software or anyone's use of the Software, this NTUITIVE-LA (and your license and rights obtained herein) terminate automatically. Upon any such termination, you shall destroy all of your copies of the Software immediately. Sections 3, 4, 5, 6, 7, 8, 11 and 12 of this NTUITIVE-LA shall survive any termination of this NTUITIVE-LA.
% 9. That the patent rights, if any, granted to you in this NTUITIVE-LA only apply to the Software, not to any derivative works you make.
% 10. That the Software may be subject to U.S. export jurisdiction at the time it is licensed to you, and it may be subject to additional export or import laws in other places. You agree to comply with all such laws and regulations that may apply to the Software after delivery of the software to you.
% 11. That all rights not expressly granted to you in this NTUITIVE-LA are reserved.
% 12. That this NTUITIVE-LA shall be construed and controlled by the laws of the Republic of Singapore without regard to conflicts of law. If any provision of this NTUITIVE-LA shall be deemed unenforceable or contrary to law, the rest of this NTUITIVE-LA shall remain in full effect and interpreted in an enforceable manner that most nearly captures the intent of the original language.
%

```
 613 %
 614 % Do you accept all of the terms of the preceding NTUITIVE-LA license agreement? If you accept the terms, click “I Agree,” then “Next.”  Otherwise click “Cancel.”
 615 %
 616 % Copyright (c) NTUITIVE. All rights reserved.
 617
 618
 619
 620 % copy of FUNWAVE's BDS license agreement:
 621 % !    Copyright (c) 2016, FUNWAVE Development Team
 622 % !
 623 % !    (See http://www.udel.edu/kirby/programs/funwave/funwave.html
 624 % !     for Development Team membership)
 625 % !
 626 % !    All rights reserved.
 627 % !
 628 % !    FUNWAVE_TVD is free software: you can redistribute it and/or modify
 629 % !    it under the terms of the Simplified BSD License as released by
 630 % !    the Berkeley Software Distribution (BSD).
 631 % !
 632 % !    Redistribution and use in source and binary forms, with or without
 633 % !    modification, are permitted provided that the following conditions are met:
 634 % !
 635 % !    1. Redistributions of source code must retain the above copyright notice, this
 636 % !       list of conditions and the following disclaimer.
 637 % !    2. Redistributions in binary form must reproduce the above copyright notice,
 638 % !    this list of conditions and the following disclaimer in the documentation
 639 % !    and/or other materials provided with the distribution.
 640 % !
 641 % !    THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
 642 % !    ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
 643 % !    WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 644 % !    DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR
 645 % !    ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
 646 % !    (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 647 % !    LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
 648 % !    ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 649 % !    (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
 650 % !    SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 651 % !
 652 % !    The views and conclusions contained in the software and documentation are those
 653 % !    of the authors and should not be interpreted as representing official policies,
 654 % !    either expressed or implied, of the FreeBSD Project.
```

```
655
656
657
658 %This is the start of the app's code:
659
660
661             %-----------------------------------------
662             %------Remove old Makefile, funwave.exe-----
663             %-----------------------------------------
664             %In reclicking the button, delete existing Makefile and funwave.↙
exe
665             if exist(fullfile(app.directory, "funwave"),'file')
666                 delete(fullfile(app.directory, "funwave"));
667             end
668
669             if exist(fullfile(app.directory,"Makefile"),'file')
670                 delete(fullfile(app.directory,"Makefile"));
671             end
672
673             %Disable the Makefile  buttons
674             app.CreateMakefileButton.Enable = 'off';
675             app.StartsimulationButton.Enable = 'off';
676
677             %Remove preview in the Makefile Tab
678             app.Preview_Makefile.Value = '';
679
680             %----------------------------
681             %------Output Directory------
682             %----------------------------
683             %Check if the output directory has been manually set by the user
684             if  app.SavefilestoEditField.Value == string(app.WORKFOLDER)
685                 inputtxt_directory =  fullfile(app.WORKFOLDER);
686                 app.SavefilestoEditField.FontColor = 'k';
687
688                 %Check if the 'INPUT_FILES' directory exists, and create it if↙
it doesn't
689                 if ~exist(inputtxt_directory, 'dir')
690                     mkdir(inputtxt_directory); % Creates the directory
691                 end
692             else
693                 %Set the default directory to the Desktop
694                 if ismac
695                     %macOS
696                     defaultDir = fullfile(getenv('HOME'), 'Desktop');
697                 elseif ispc
698                     %Windows
699                     defaultDir = fullfile(getenv('USERPROFILE'), 'Desktop');
700                 else
701                     %Handle other platforms
702                     defaultDir = pwd;
703                 end
704
705                     Dir1 = fullfile(defaultDir, 'INPUT_FILES'); %Create↙
'INPUT_FILES' folder
706
707                     % Check if the 'INPUT_FILES' directory exists, and create↙
it if it doesn't
708                     if ~exist(Dir1, 'dir')
```

```
709                         mkdir(Dir1); % Creates the directory
710                     end
711
712                     app.SavefilestoEditField.Value = deblank(string(Dir1)); %"Current Folder";
713                     inputtxt_directory =  deblank(string(Dir1));
714                     app.SavefilestoEditField.FontColor = 'k';
715             end
716
717             app.directory  = inputtxt_directory;
718
719
720             %-----------------------------------
721             %-----CREATE INPUT.TXT HEADER------
722             %-----------------------------------
723             %Create the input.txt
724             fileInputPath = fullfile(inputtxt_directory, 'input.txt');
725             fileInputId = fopen(fileInputPath, 'w');
726
727             %Template of the first few lines of the input.txt
728             templateStr = [
729                 '! INPUT FILE FOR BOUSS_TVD\n' ...
730                 '! Note: all input parameter are capital sensitive\n\n' ...
731                 '! Generated using FUNMAP by Felix, et al. (2025) \n\n'...
732                 '! --------------------TITLE-------------------------------------\n' ...
733                 'TITLE = FUNWAVE\n\n' ...
734                 '  ! -------------------HOTSTART---------------------------------\n' ...
735                 'HOT_START = F\n' ...
736                 'FileNumber_HOTSTART = HotStartMTLB\n'];
737
738             %Add the template into the text file
739             fprintf(fileInputId, templateStr);
740
741             %Create Input_Data subfolder
742             inputFolder = fullfile(inputtxt_directory, 'Input_Data');
743             app.inputFolder = inputFolder;
744
745             %Check if the 'INPUT_FILES/Input_Data' directory exists, create it if it doesn't
746             if ~exist(app.inputFolder, 'dir')
747                 mkdir(app.inputFolder);  % Creates the 'INPUTFOLDER' directory
748             end
749
750
751             %-----------------------
752             %------LOG REPORT------
753             %-----------------------
754             %Create 'Log_Files' subfolder
755             LogFolder = fullfile(inputtxt_directory, 'Log_Files');
756             LogFile = fullfile(LogFolder,'Log_Report_InputPrep.txt');
757
758                 % Check if the 'INPUT_FILES/Log_Files' directory exists, create it if it doesn't
759                 if ~exist(LogFolder, 'dir')
760                     mkdir(LogFolder);
761                 end
```

```
 762
 763                 %Create the log text file
 764                 fileId = fopen(fullfile(LogFolder,'Log_Report_InputPrep.txt'),'w');
 765
 766                 %Template for the default section of the log report
 767                 fprintf(fileId, '%s\n','******************** LOG REPORT ******************** ');
 768                 fprintf(fileId, '%-30s%-s\n', 'Type:', 'input.txt, Makefile, funwave executable');
 769                 timestamp = datestr(now, 'yyyy-mm-dd HH:MM:SS'); % Get the current timestamp
 770                 fprintf(fileId, '%-30s%-s\n','Timestamp:', timestamp); %Add timestamp in the header
 771                 fprintf(fileId, '%s\n','************************************************************');
 772                 fprintf(fileId,'%-s\n','');
 773                 fprintf(fileId,'%-s\n\n','*******  INPUT.TXT SECTION *******');
 774
 775                 error = 0; % General error counter
 776                 error_mkfile = 0; %Makefile error counter
 777
 778                 %-------------------------
 779                 %------PARALLEL INFO------
 780                 %-------------------------
 781                 tempParallel = sprintf(['\n! -------------------PARALLEL INFO----------------------------\n' ...
 782                     '!    PX,PY - processor numbers in X and Y\n' ...
 783                     '!    NOTE: make sure consistency with mpirun -np n (px*py)\n' ...
 784                     'PX = %.0f\n' ...
 785                     'PY = %.0f\n'], ...
 786                     app.ProcessorX.Value, ...
 787                     app.ProcessorY.Value);
 788
 789                 %Add these lines in the input.txt
 790                 fprintf(fileInputId, '%s', tempParallel);
 791
 792                 %-----------------
 793                 %------DEPTH------
 794                 %-----------------
 795                 %If no option is selected
 796                 if app.Button4.Value
 797                     app.ImportDataButton.FontColor = 'r';
 798                     app.FlatButton.FontColor = 'r';
 799                     app.SlopeButton.FontColor = 'r';
 800                     fprintf(fileId,'%-30s%-s\n','DEPTH:', 'No data is selected');
 801                     error = error + 1; %to note an error occurred
 802                 end
 803
 804                 %------------------------------
 805                 %------DEPTH: Import Data------
 806                 %------------------------------
 807                 if app.ImportDataButton.Value
 808                     % Check if no file is uploaded
 809                     Depthconditions = [...
```

```
 810                         isempty(app.DepthFileEditField.Value), ...
 811                         ~strcmp(app.DepthFileEditField.Value, app.DEPTHFILE) ↙
...
 812                         str2double(app.DepthFileEditField.Value) == 0 ...
 813                         ];
 814
 815                     %Display 'NO FILE' in the textbox and note the issue in ↙
the log report
 816                     if any(Depthconditions)
 817                         app.DepthFileEditField.BackgroundColor = 'r';
 818                         app.DepthFileEditField.Value = 'NO FILE';
 819                         app.DepthFileEditField.FontColor = 'w';
 820                         fprintf(fileId,'%-30s%-s\n','DEPTH:', 'No file is ↙
uploaded');
 821                         error = error + 1;
 822
 823                     else %Read the bathymetry file
 824                         if app.InvertvaluesCheckBox.Value
 825                             %Convert water depth values to positive values
 826                             depth = -app.depthdata;
 827                         else
 828                             depth = app.depthdata;
 829                         end
 830
 831                         % Round off depth to the nearest five decimal places
 832                         if isnumeric(depth) && isscalar(depth)
 833                             %Scalar numeric input
 834                             app.depthfinal = round(depth, 5);
 835
 836                         elseif ischar(depth) || isstring(depth)
 837                             %String input
 838                             depthNum = str2double(depth);
 839                             if ~isnan(depthNum)
 840                                 app.depthfinal = round(depthNum, 5);
 841                             end
 842                         elseif isnumeric(depth) && ~isscalar(depth)
 843                             %Numeric array inpupt
 844                             app.depthfinal = round(double(depth), 5);
 845                         else
 846                             app.depthfinal = round(depth, 5);
 847                         end
 848
 849                         %Save the bathymetry file in 'INPUT_FILES/Input_Data/'
 850                         outputFilePath = fullfile(app.inputFolder,app. ↙
DEPTHFILE);
 851
 852
 853                         % Open the file for writing
 854                         fileID0 = fopen(outputFilePath, 'w');
 855
 856                         % Define the format for each row (e.g., '%.5f' for two ↙
decimal places)
 857                         formatSpec = [repmat('%.5f\t', 1, size(app.depthfinal, ↙
2) - 1), '%.5f\n'];
 858
 859                         % Write each row of depthfinal to the file
 860                         for row = 1:size(app.depthfinal, 1)
 861                             fprintf(fileID0, formatSpec, app.depthfinal(row, ↙
```

```
:));
 862                         end
 863
 864                         % Close the file
 865                         fclose(fileID0);
 866
 867
 868
 869                         tempDepth = sprintf([
 870                             '\n  ! --------------------↙
DEPTH-------------------------------------\n' ...
 871                             'DEPTH_TYPE = DATA\n' ...
 872                             'DEPTH_FILE =  Input_Data/%s\n' ...
 873                             'DepthFormat = ELE\n' ...
 874                             'WaterLevel = %.1f\n'], ...
 875                             app.DEPTHFILE,app.WaterlevelmEditField.Value);
 876                     end
 877                 end
 878
 879                 %-----------------------
 880                 %------DEPTH: Flat------
 881                 %-----------------------
 882                 if app.FlatButton.Value
 883                     tempDepth = sprintf([
 884                         '\n  ! --------------------↙
DEPTH-------------------------------------\n' ...
 885                         'DEPTH_TYPE = FLAT\n' ...
 886                         'DEPTH_FLAT = %.2f\n'], ...
 887                         app.DepthmEditField.Value);
 888
 889                     % Add warning note in the log report if the 0 value is↙
unchanged
 890                     if app.DepthmEditField.Value == 0
 891                         app.DepthmEditField.FontColor = 'r';
 892                         fprintf(fileId,'%-30s%-s\n','DEPTH:', 'Flat depth is↙
zero');
 893                         error = error + 1;
 894                     end
 895                 end
 896
 897                 %-----------------------
 898                 %------DEPTH: Slope------
 899                 %-----------------------
 900                 if app.SlopeButton.Value
 901                     tempDepth = sprintf([
 902                         '\n  ! --------------------↙
DEPTH-------------------------------------\n' ...
 903                         'DEPTH_TYPE = SLOPE\n' ...
 904                         'DEPTH_FLAT = %.2f\n' ...
 905                         'SLP = %.2f\n' ...
 906                         'Xslp = %.2f\n'], ...
 907                         app.DepthmEditField.Value, ...
 908                         app.AngleEditField.Value, ...
 909                         app.StartatXEditField.Value);
 910
 911
 912                     %If one of the depth options is selcted
 913                     if exist('tempDepth', 'var') == 1
```

```
914                         fprintf(fileInputId, '%s', tempDepth); %Add the Depth info in the input.txt
915                     end
916
917                     % Add warning note in the log report if 0 value is unchanged
918                     if app.DepthmEditField.Value == 0
919                         app.DepthmEditField.FontColor = 'r';
920                         fprintf(fileId,'%-30s%-s\n','DEPTH:', 'Flat depth is zero');
921                         error = error + 1;
922                     end
923
924                     if app.DepthmEditField.Value == 0
925                         app.DepthmEditField.FontColor = 'r';
926                         fprintf(fileId,'%-30s%-s\n','DEPTH:', 'Slope depth is zero');
927                         error = error + 1;
928                     end
929
930                     if app.AngleEditField.Value == 0
931                         app.AngleEditField.FontColor = 'r';
932                         fprintf(fileId,'%-30s%-s\n','DEPTH:', 'Slope angle is zero');
933                         error = error + 1;
934                     end
935
936                 end
937
938                 if exist("tempDepth",'var')
939                     fprintf(fileInputId, '%s', tempDepth);
940                 end
941
942                 %---------------------
943                 %------DIMENSION------
944                 %---------------------
945                 tempDim = sprintf([
946                     '\n  ! ------------------DIMENSION-----------------------------\n' ...
947                     '  ! global grid dimension\n' ...
948                     'Mglob = %d\n' ...
949                     'Nglob = %d\n'], ...
950                     app.XEditField_2.Value, ...
951                     app.YEditField_2.Value);
952
953                 %Add these lines in the input.txt
954                 fprintf(fileInputId, '%s', tempDim);
955
956                 %Add warning note in the log report if the 0 value is unchanged
957                 if app.XEditField_2.Value == 0
958                     app.XEditField_2.FontColor = 'r';
959                     fprintf(fileId,'%-30s%-s\n','DIMENSION:', 'X is zero');
960                     error = error + 1;
961                 end
962
963                 if app.YEditField_2.Value == 0
964                     app.YEditField_2.FontColor = 'r';
```

```
 965                     fprintf(fileId,'%-30s%-s\n','DIMENSION:', 'Y is zero');
 966                     error = error + 1;
 967                 end
 968
 969                 %-----------------
 970                 %------PRINT------
 971                 %-----------------
 972                 % Identify the correct directory separator based on the operating system
 973                 if ismac
 974                     dirSeparator = "/";
 975                 elseif ispc
 976                     dirSeparator = "\";
 977                 else
 978                     dirSeparator = "/";
 979                 end
 980
 981                 if isempty(app.ResultsfolderEditField.Value)
 982                     %Use 'Simulation_Results' as the default folder name
 983                     RESULT_FOLDER = "Simulation_Results" + dirSeparator;
 984                     app.ResultsfolderEditField.Value = RESULT_FOLDER;
 985                     app.ResultsfolderEditField.FontColor = 'k';
 986                 else %Use the user's input
 987                     % if endsWith(app.ResultsfolderEditField.Value, dirSeparator)
 988                         RESULT_FOLDER = app.ResultsfolderEditField.Value;
 989                     % else
 990                     %     RESULT_FOLDER = app.ResultsfolderEditField.Value + dirSeparator;
 991                     %     app.ResultsfolderEditField.Value = RESULT_FOLDER;
 992                     % end
 993                 end
 994
 995                 tempPrint = sprintf([
 996                     '\n  ! -------------------PRINT---------------------------------\n' ...
 997                     'RESULT_FOLDER = %s\n'], ...
 998                     RESULT_FOLDER); %
 999
1000                 %Add these lines in the input.txt
1001                 fprintf(fileInputId, '%s', tempPrint);
1002
1003                 %----------------
1004                 %------TIME------
1005                 %----------------
1006                 tempTime = sprintf([
1007                     '\n  ! -----------------TIME---------------------------------\n' ...
1008                     '  ! time: total computational time / plot time / screen interval \n' ...
1009                     '  ! all in seconds\n' ...
1010                     'PLOT_START_TIME = %.1f\n' ...
1011                     'TOTAL_TIME = %.1f\n' ...
1012                     'PLOT_INTV = %.1f\n' ...
1013                     'PLOT_INTV_STATION = %.1f\n' ...
1014                     'SCREEN_INTV = %.1f\n'], ...
1015                     app.StartEditField.Value, ...
1016                     app.TotalEditField.Value, ...
```

```
1017                        app.SimulationIntervalEditField.Value, ...
1018                        app.GaugeIntervalEditField.Value, ...
1019                        app.SimulationIntervalEditField.Value);
1020
1021                    %Add these lines in the input.txt
1022                    fprintf(fileInputId, '%s', tempTime);
1023
1024                    %----------------
1025                    %------GRID------
1026                    %----------------
1027                    tempGrid = sprintf([
1028                        '\n  ! -----------------↙
GRID----------------------------------\n' ...
1029                        'StretchGrid = F\n']);
1030
1031                    fprintf(fileInputId, '%s', tempGrid);
1032
1033                    %---------------------------
1034                    %------GRID: Spherical------
1035                    %---------------------------
1036                    %Changing the parameter variable for Grid Size based on the↙
coordinate system
1037                    if app.SphericalButton.Value
1038                        tempGrid2 = sprintf([
1039                            '!spherical grid, in decimal degrees\n' ...
1040                            'Lon_West = %.4f\n' ...
1041                            'Lat_South = %.4f\n' ...
1042                            'Dphi = %.4f\n' ...
1043                            'Dtheta = %.4f\n'], ...
1044                            app.LongitudeEditField.Value, app.LatitudeEditField.↙
Value, app.XEditField.Value, app.YEditField.Value);
1045
1046                        %Add warning note in the log report if the 0 value is↙
unchanged
1047                        if app.LongitudeEditField.Value == 0
1048                            app.LongitudeEditField.FontColor = 'r';
1049                            fprintf(fileId,'%-30s%-s\n','GRID:', 'Longitude is↙
zero');
1050                            error = error + 1;
1051                        end
1052                        if app.LatitudeEditField.Value == 0
1053                            app.LatitudeEditField.FontColor = 'r';
1054                            fprintf(fileId,'%-30s%-s\n','GRID:', 'Latitude is↙
zero');
1055                            error = error + 1;
1056                        end
1057
1058                        if app.LatitudeEditField.Value == 0 && app.↙
LongitudeEditField.Value == 0
1059                            error_mkfile = error_mkfile +1;
1060                            error_mkfiletext3 = sprintf('Grid');
1061                        end
1062
1063                        if app.XEditField.Value == 0
1064                            app.XEditField.FontColor = 'r';
1065                            fprintf(fileId,'%-30s%-s\n','GRID:', 'X is zero');
1066                            error = error + 1;
1067                        end
```

```
1068                        if app.YEditField.Value == 0
1069                            app.YEditField.FontColor = 'r';
1070                            fprintf(fileId,'%-30s%-s\n','GRID:', 'Y is zero');
1071                            error = error + 1;
1072                        end
1073
1074                        if app.XEditField.Value == 0 && app.YEditField.Value == 0
1075                            error_mkfile = error_mkfile +1;
1076                            error_mkfiletext3 = sprintf('Grid');
1077                        end
1078
1079                        if app.ArrivalCheckBox.Value
1080                            % Include the arrival time threshold
1081                            tempGrid2 = [tempGrid2, sprintf('ArrTimeMinH = %.↙
4f\n', app.ThresholdEditField.Value)];
1082                        end
1083
1084                    %---------------------------
1085                    %------GRID: Cartesian------
1086                    %---------------------------
1087                    elseif app.CartesianButton.Value
1088                        tempGrid2 = sprintf([
1089                            '! cartesian grid sizes\n' ...
1090                            'DX = %.2f\n' ...
1091                            'DY = %.2f\n'], ...
1092                            app.XEditField.Value, app.YEditField.Value);
1093
1094                        app.LatitudeEditField.FontColor = '0.65,0.65,0.65';
1095                        app.LongitudeEditField.FontColor = '0.65,0.65,0.65';
1096
1097                        if app.XEditField.Value == 0
1098                            app.XEditField.FontColor = 'r';
1099                            fprintf(fileId,'%-30s%-s\n','GRID:', 'X is zero');
1100                            error = error + 1;
1101                        end
1102                        if app.YEditField.Value == 0
1103                            app.YEditField.FontColor = 'r';
1104                            fprintf(fileId,'%-30s%-s\n','GRID:', 'Y is zero');
1105                            error = error + 1;
1106                        end
1107
1108                        if app.XEditField.Value == 0 && app.YEditField.Value == 0
1109                            error_mkfile = error_mkfile +1;
1110                            error_mkfiletext3 = sprintf('Grid');
1111                        end
1112
1113                    %----------------------
1114                    %------GRID: None------
1115                    %----------------------
1116                    elseif app.Button_2.Value
1117                        %If no Cooordinate system is selected, add an error in the↙
log report
1118                        app.SphericalButton.FontColor = 'r';
1119                        app.CartesianButton.FontColor = 'r';
1120                        fprintf(fileId,'%-30s%-s\n','GRID:', 'No coordinate system↙
is selected ');
1121                        tempGrid2 = '';
1122                        error = error + 1;
```

```
1123                        error_mkfile = error_mkfile +1;
1124                        error_mkfiletext3 = sprintf('Grid');
1125                    end
1126
1127                    %Add these lines in the input.txt
1128                    fprintf(fileInputId, '%s', tempGrid2);
1129
1130                    %----------------------------
1131                    %------INITIAL CONDITION------
1132                    %----------------------------
1133                    if app.Button_31.Value
1134                        fprintf(fileId,'%-30s%-s\n','INITIAL CONDITION:', 'None selected. Set it up in Tab 2');
1135                        error = error + 1;
1136                    end
1137
1138                    %----------------------------------------
1139                    %------INITIAL CONDITION: Import UVZ------
1140                    %----------------------------------------
1141                    if app.ImportUVZButton.Value
1142                        tempUVZ1 = sprintf([
1143                            '\n  ! --------------- INITIAL UVZ ----------------------------\n' ...
1144                            '  ! INI_UVZ - initial UVZ e.g., initial deformation\n' ...
1145                            '  !         must provide three (3) files \n' ...
1146                            'INI_UVZ = T\n']);
1147
1148                        %Set up the initial values for the three parameters to avoid errors when creating the text file
1149                        tempUVZ2 = sprintf('ETA_FILE = \n');
1150                        tempUVZ3 = sprintf('U_FILE = \n');
1151                        tempUVZ4 = sprintf('V_FILE = \n');
1152
1153                        %Check if the eta file / Z file is loaded
1154                        ETAconditions = [...
1155                            isempty(app.SurfaceHeightEditField.Value), ...
1156                            ~any(strcmp(app.SurfaceHeightEditField.Value, app.SurfaceHeightNAME)), ...
1157                            str2double(app.SurfaceHeightEditField.Value) == 0  ...
1158                            ];
1159
1160                        if any(ETAconditions)
1161                            %Display 'NO FILE' in the textbox and note the issue in the log report
1162                            fprintf(fileId, '%-30s%-s\n', 'INITIAL CONDITION:', 'Import UVZ:');
1163                            fprintf(fileId, '%-30s%-s\n', '', 'No file uploaded in ''Surface Height'''');
1164                            app.SurfaceHeightEditField.Value = "NO FILE";
1165                            app.SurfaceHeightEditField.FontColor = 'w';
1166                            app.SurfaceHeightEditField.BackgroundColor = 'r';
1167                            error = error + 1;
1168                        elseif strcmp(app.SurfaceHeightEditField.Value, app.SurfaceHeightNAME)
1169                           %If the filename matches the name in the textbox, add it to input.txt
1170                            tempUVZ2 = sprintf('ETA FILE = Input Data/%s\n', app.
```

```
SurfaceHeightNAME);
1171
1172                         %If the file is not in 'INPUT_FILES/Input_Data', copy↙
it there
1173                         if ~exist(fullfile(app.inputFolder, app.↙
SurfaceHeightNAME), 'file')
1174                             copyfile(app.SurfaceHeightLOC, app.inputFolder);
1175                         end
1176                     end
1177
1178                     % Checking if U vector file is loaded
1179                     uConditions = [...
1180                         isempty(app.UVelocityEditField.Value), ... % Check if↙
the field is empty
1181                         ~any(strcmp(app.UVelocityEditField.Value, app.↙
XVelocityNAME)), ... % Check if the value does not match expected names
1182                         str2double(app.UVelocityEditField.Value) == 0 ... %↙
Check if the value is zero (assuming numeric comparison is relevant)
1183                         ];
1184
1185                     if any(uConditions)
1186                         %Display 'NO FILE' in the textbox and note the issue↙
in the log report
1187                         fprintf(fileId, '%-30s%-s\n', 'INITIAL CONDITION:',↙
'Import UVZ:');
1188                         fprintf(fileId, '%-30s%-s\n', '', 'No file uploaded in↙
''U Velocity''');
1189                         app.UVelocityEditField.Value = "NO FILE";
1190                         app.UVelocityEditField.FontColor = 'w';
1191                         app.UVelocityEditField.BackgroundColor = 'r';
1192                         error = error + 1;
1193                     else
1194                         %If the filename matches the name in the textbox, add↙
it to input.txt
1195                         tempUVZ3 = sprintf('U_FILE = Input_Data/%s\n', app.↙
XVelocityNAME);
1196
1197                         %If the file is not in 'INPUT_FILES/Input_Data', copy↙
it there
1198                         if ~exist(fullfile(app.inputFolder, app.↙
XVelocityNAME), 'file')
1199                             copyfile(app.XVelocityLOC, app.inputFolder);
1200                         end
1201                     end
1202
1203                     % Checking if V vector file is loaded
1204                     vConditions = [...
1205                         isempty(app.VVelocityEditField.Value), ... % Check if↙
the field is empty
1206                         ~any(strcmp(app.VVelocityEditField.Value, app.↙
YVelocityNAME)), ... % Check if the value does not match expected names
1207                         str2double(app.VVelocityEditField.Value) == 0 ... %↙
Check if the value is zero (assuming numeric comparison is relevant)
1208                         ];
1209
1210                     if any(vConditions)
1211                         %Display 'NO FILE' in the textbox and note the issue↙
in the log report
```

```
1212                        fprintf(fileId, '%-30s%-s\n', 'INITIAL CONDITION:',↙
'Import UVZ:');
1213                        fprintf(fileId, '%-30s%-s\n', '', 'No file uploaded in↙
''V Velocity'''');
1214                        app.VVelocityEditField.Value = "NO FILE";
1215                        app.VVelocityEditField.FontColor = 'w';
1216                        app.VVelocityEditField.BackgroundColor = 'r';
1217                        error = error + 1;
1218                    else
1219                        %If the filename matches the name in the textbox, add↙
it to input.txt
1220                        tempUVZ4 = sprintf('V_FILE = Input_Data/%s\n', app.↙
YVelocityNAME);
1221
1222                        %If the file is not in 'INPUT_FILES/Input_Data', copy↙
it there
1223                        if exist(fullfile(inputFolder, app.YVelocityNAME),↙
'file') == 2
1224                            %do nothing
1225                        else
1226                            %Copy the file to the OUTPUT_FOLDER
1227                            copyfile(app.YVelocityLOC, inputFolder);
1228                        end
1229                    end
1230
1231                    % Combine all the the parts
1232                    tempUVZ = [tempUVZ1,tempUVZ2,tempUVZ3,tempUVZ4];
1233
1234                    %Add these lines in the input.txt
1235                    fprintf(fileInputId, '%s', tempUVZ);
1236                else
1237                    fprintf(fileInputId, [
1238                        '\n  ! --------------- INITIAL UVZ↙
--------------------------\n' ...
1239                        'INI_UVZ = F\n']);
1240                end
1241
1242                %---------------------------------------
1243                %------INITIAL CONDITION: Wavemaker------
1244                %---------------------------------------
1245                %To list all the errors in the log report
1246            function checkAndReport(field, fieldName, fileId)
1247                if field.Value == 0
1248                    field.FontColor = 'r';
1249                    fprintf(fileId, '%-30s%-s\n', 'INITIAL CONDITION:',↙
'Wavemaker:');
1250                    fprintf(fileId, '%-35s%-s\n', '', [fieldName ' is zero']);
1251                    error = error + 1;
1252                else
1253                    field.FontColor = 'k';
1254                end
1255            end
1256
1257            if app.WavemakerButton.Value
1258                %Define variables
1259                Xc = app.XcoordinatemEditField.Value;
1260                Yc = app.YcoordinatemEditField.Value;
1261                WID = app.WidthmEditField.Value;
```

```
1262                    AMP = app.AmplitudemEditField.Value;
1263                    DEP = app.WaterdepthmEditField.Value;
1264                    LAGTIME = app.LagtimesecEditField.Value;
1265                    Tperiod = app.PeriodsecEditField.Value;
1266                    Theta_WK = app.ThetadegreesEditField.Value;
1267                    Time_ramp = app.TimerampsecEditField.Value;
1268                    NumWaveComp = app.WaveCompCountEditField.Value;
1269                    FreqPeak =  app.PeakEditField.Value;
1270                    FreqMin = app.MinimumEditField.Value;
1271                    FreqMax = app.MaximumEditField.Value;
1272                    Hmo = app.WaveHeightmEditField.Value;
1273
1274
1275                    tempWavemaker1 = sprintf(['\n! ----------------↙
WAVEMAKER------------------------------\n']);
1276
1277                    %Add only the necessary parameters based on the dropdown↙
selection
1278                    switch app.WavemakertypeDropDown.Value
1279
1280                        case "INI_REC"
1281                            %-------------------
1282                            %......INI_REC......
1283                            %-------------------
1284                            tempWavemaker2 = sprintf('WAVEMAKER = INI_REC \nXc =↙
%.4f\nYc = %.4f\nWID = %.2f\n', Xc, Yc, WID);
1285
1286                            %Log report
1287                            checkAndReport(app.XcoordinatemEditField, 'X↙
Coordinate', fileId);
1288                            checkAndReport(app.YcoordinatemEditField, 'Y↙
Coordinate', fileId);
1289                            checkAndReport(app.WidthmEditField, 'Width (m)',↙
fileId);
1290
1291
1292                        case "JON_1D"
1293                            %------------------
1294                            %......JON_1D......
1295                            %------------------
1296                            tempWavemaker2 = sprintf('WAVEMAKER = JON_1D\nXc_WK =↙
%.4f\nYc_WK = %.4f\nYwidth_WK = %.2f\nDEP_WK = %.2f\nTime_ramp = %.2f\nDelta_WK =↙
%.2f\nFreqPeak = %.2f\nFreqMin = %.2f\nFreqMax = %.2f\nHmo = %.2f\nGammaTMA = 3.3↙
\nNfreq = 45.0\n', Xc, Yc, WID,DEP, Time_ramp,app.WidthDeltaEditField.Value,↙
FreqPeak,FreqMin,FreqMax,Hmo);
1297
1298                            %Log report
1299                            checkAndReport(app.XcoordinatemEditField, 'X↙
Coordinate', fileId);
1300                            checkAndReport(app.YcoordinatemEditField, 'Y↙
Coordinate', fileId);
1301                            checkAndReport(app.WidthmEditField, 'Width (m)',↙
fileId);
1302                            checkAndReport(app.WaterdepthmEditField, 'Water depth↙
(m)', fileId);
1303                            checkAndReport(app.TimerampsecEditField, 'Time ramp↙
(sec)', fileId);
1304                            checkAndReport(app.WidthDeltaEditField, 'Width Delta',↙
```

```
fileId);
                        checkAndReport(app.MinimumEditField, 'Frequency Minimum', fileId);
                        checkAndReport(app.MaximumEditField, 'Frequency Maximum', fileId);
                        checkAndReport(app.PeakEditField, 'Frequency Peak', fileId);
                        checkAndReport(app.WaveHeightmEditField, 'Wave height (hmo)', fileId);

                    case "JON_2D"
                        %------------------
                        %......JON_2D......
                        %------------------
                        tempWavemaker2 = sprintf('WAVEMAKER = JON_2D\nXc_WK = %.4f\nYc_WK = %.4f\nYwidth_WK = %.2f\nDEP_WK = %.2f\nTime_ramp = %.2f\nDelta_WK = %.2f\nFreqPeak = %.2f\nFreqMin = %.2f\nFreqMax = %.2f\nHmo = %.2f\nGammaTMA = 3.3\nThetaPeak = 0.0\nNfreq = 45.0\nNtheta = 24.0\n', Xc, Yc, WID,DEP, Time_ramp,app.WidthDeltaEditField.Value, FreqPeak,FreqMin,FreqMax,Hmo);

                        %Log report
                        checkAndReport(app.XcoordinatemEditField, 'X Coordinate', fileId);
                        checkAndReport(app.YcoordinatemEditField, 'Y Coordinate', fileId);
                        checkAndReport(app.WidthmEditField, 'Width (m)', fileId);
                        checkAndReport(app.WaterdepthmEditField, 'Water depth (m)', fileId);
                        checkAndReport(app.TimerampsecEditField, 'Time ramp (sec)', fileId);
                        checkAndReport(app.WidthDeltaEditField, 'Width Delta', fileId);
                        checkAndReport(app.MinimumEditField, 'Frequency Minimum', fileId);
                        checkAndReport(app.MaximumEditField, 'Frequency Maximum', fileId);
                        checkAndReport(app.PeakEditField, 'Frequency Peak', fileId);
                        checkAndReport(app.WaveHeightmEditField, 'Wave height (hmo)', fileId);

                    case "TMA_1D"
                        %------------------
                        %......TMA_1D......
                        %------------------
                        tempWavemaker2 = sprintf('WAVEMAKER = TMA_1D\nXc_WK = %.4f\nYc_WK = %.4f\nYwidth_WK = %.2f\nDEP_WK = %.2f\nTime_ramp = %.2f\nDelta_WK = %.2f\nFreqPeak = %.2f\nFreqMin = %.2f\nFreqMax = %.2f\nHmo = %.2f\nGammaTMA = 3.3\nNfreq = 45.0\n', Xc, Yc, WID,DEP, Time_ramp,app.WidthDeltaEditField.Value, FreqPeak,FreqMin,FreqMax,Hmo);

                        %Log report
                        checkAndReport(app.XcoordinatemEditField, 'X Coordinate', fileId);
                        checkAndReport(app.YcoordinatemEditField, 'Y Coordinate', fileId);
                        checkAndReport(app.WidthmEditField, 'Width (m)',
```

```
fileId);
                        checkAndReport(app.WaterdepthmEditField, 'Water depth (m)', fileId);
                        checkAndReport(app.TimerampsecEditField, 'Time ramp (sec)', fileId);
                        checkAndReport(app.WidthDeltaEditField, 'Width Delta', fileId);
                        checkAndReport(app.MinimumEditField, 'Frequency Minimum', fileId);
                        checkAndReport(app.MaximumEditField, 'Frequency Maximum', fileId);
                        checkAndReport(app.PeakEditField, 'Frequency Peak', fileId);
                        checkAndReport(app.WaveHeightmEditField, 'Wave height (hmo)', fileId);

                    case "LEF_SOL"
                        %-------------------
                        %......LEF_SOL......
                        %-------------------
                        tempWavemaker2 = sprintf('WAVEMAKER = LEF_SOL\nAMP = %.2f\nDEP = %.2f\nLAGTIME = %.2f\n', AMP, DEP, LAGTIME);

                        %Log report
                        checkAndReport(app.AmplitudemEditField, 'Amplitude (m)', fileId);
                        checkAndReport(app.WaterdepthmEditField, 'Water depth (m)', fileId);
                        checkAndReport(app.LagtimesecEditField, 'Lag time (sec)', fileId);

                    case "INI_SOL"
                        %-------------------
                        %......INI_SOL......
                        %-------------------
                        tempWavemaker2 = sprintf('WAVEMAKER = INI_SOL\nAMP = %.2f\nDEP = %.2f\nXWAVEMAKER = %.4f\n', AMP, DEP, Xc);

                        %Log report
                        checkAndReport(app.AmplitudemEditField, 'Amplitude (m)', fileId);
                        checkAndReport(app.WaterdepthmEditField, 'Water depth (m)', fileId);
                        checkAndReport(app.XcoordinatemEditField, 'X Coordinate', fileId);

                    case "WK_IRR"
                        %-----------------
                        %......WK_IRR......
                        %-----------------
                        tempWavemaker2 = sprintf('WAVEMAKER = WK_IRR\nXc_WK = %.4f\nYc_WK = %.4f\nYwidth_WK = %.2f\nDEP_WK = %.2f\nTime_ramp = %.2f\nDelta_WK = %.2f\nFreqPeak = %.2f\nFreqMin = %.2f\nFreqMax = %.2f\nHmo = %.2f\nGammaTMA = 3.3 \nThetaPeak = 0.0\nNfreq = 45.0\nNtheta = 24.0\nEqualEnergy = TRUE', Xc, Yc, WID, DEP, Time_ramp,app.WidthDeltaEditField.Value, FreqPeak,FreqMin,FreqMax,Hmo);

                        %Log report
                        checkAndReport(app.XcoordinatemEditField, 'X
```

```
Coordinate', fileId);
1376                         checkAndReport(app.YcoordinatemEditField, 'Y↙
Coordinate', fileId);
1377                         checkAndReport(app.WidthmEditField, 'Width (m)',↙
fileId);
1378                         checkAndReport(app.WaterdepthmEditField, 'Water depth↙
(m)', fileId);
1379                         checkAndReport(app.TimerampsecEditField, 'Time ramp↙
(sec)', fileId);
1380                         checkAndReport(app.WidthDeltaEditField, 'Width Delta',↙
fileId);
1381                         checkAndReport(app.MinimumEditField, 'Frequency↙
Minimum', fileId);
1382                         checkAndReport(app.MaximumEditField, 'Frequency↙
Maximum', fileId);
1383                         checkAndReport(app.PeakEditField, 'Frequency Peak',↙
fileId);
1384                         checkAndReport(app.WaveHeightmEditField, 'Wave height↙
(hmo)', fileId);
1385
1386                     case "WK_REG"
1387                         %------------------
1388                         %......WK_REG......
1389                         %------------------
1390                         tempWavemaker2 = sprintf('WAVEMAKER = WK_REG\nXc_WK =↙
%.4f\nYc_WK = %.4f\nYwidth_WK = %.2f\nTperiod = %.2f\nAMP_WK = %.2f\nDEP_WK = %.↙
2f\nTheta_WK = %.2f\nTime_ramp = %.2f\nDelta_WK = %.2f\n!The Delta_WK option is↙
activated as it is included in the simple_cases/sediment_rip file\n', Xc, Yc, WID,↙
Tperiod, AMP, DEP, Theta_WK, Time_ramp,app.WidthDeltaEditField.Value);
1391
1392                         %Log report
1393                         checkAndReport(app.XcoordinatemEditField, 'X↙
Coordinate', fileId);
1394                         checkAndReport(app.PeriodsecEditField, 'Period (sec)',↙
fileId);
1395                         checkAndReport(app.AmplitudemEditField, 'Amplitude↙
(m)', fileId);
1396                         checkAndReport(app.WaterdepthmEditField, 'Water depth↙
(m)', fileId);
1397                         checkAndReport(app.ThetadegreesEditField, 'Theta↙
(degrees)', fileId);
1398                         checkAndReport(app.TimerampsecEditField, 'Time ramp↙
(sec)', fileId);
1399
1400                     case "WK_TIME_SERIES"
1401                         %--------------------------
1402                         %......WK_TIME_SERIES......
1403                         %--------------------------
1404                         tempWavemaker2 = sprintf('WAVEMAKER = WK_TIME_SERIES\n↙
WaveCompFile = Input_Data/%s\n NumWaveComp = %.2f\nPeakPeriod = %.2f\nDEP_WK = %.↙
2f\nXc_WK = %.4f\nYwidth_WK = %.4f\n', app.WavecomponentfileEditField.Value,↙
NumWaveComp, Tperiod, DEP, Xc, WID);
1405
1406                         checkAndReport(app.WaveCompCountEditField, 'Wave↙
component count', fileId);
1407                         checkAndReport(app.PeriodsecEditField, 'Period (sec)',↙
fileId);
1408                         checkAndReport(app.WaterdepthmEditField, 'Water depth↙
```

```
(m)', fileId);
1409                            checkAndReport(app.XcoordinatemEditField, 'X↙
Coordinate', fileId);
1410                            checkAndReport(app.WidthmEditField, 'Width (m)',↙
fileId);
1411
1412                            %Check if the file is loaded
1413                            wavecompconditions = [...
1414                                isempty(app.WavecomponentfileEditField.Value ),↙
... % Check if the field is empty
1415                                ~any(strcmp(app.WavecomponentfileEditField.Value ,↙
app.WaveFileNAME)), ... % Check if the value does not match expected names
1416                                str2double(app.WavecomponentfileEditField.Value )↙
== 0 ... % Check if the value is zero (assuming numeric comparison is relevant)
1417                                ];
1418
1419                            if any(wavecompconditions)
1420                                %Display 'NO FILE' in the textbox and note the↙
issue in the log report
1421                                fprintf(fileId,'%-30s%-s\n','INITIAL↙
CONDITION:','Wavemaker:');
1422                                fprintf(fileId, '%-35s%-s\n', '', 'No file↙
uploaded in ''Wave component file''');
1423                                app.WavecomponentfileEditField.Value  = "NO FILE";
1424                                app.WavecomponentfileEditField.FontColor = 'w';
1425                                app.WavecomponentfileEditField.BackgroundColor =↙
'r';
1426                                error = error + 1;
1427                            elseif strcmp(app.WavecomponentfileEditField.Value ,↙
app.WaveFileNAME)
1428                                %If the file is not in 'INPUT_FILES/Input_Data',↙
copy it there
1429                                if ~exist(fullfile(app.inputFolder, app.↙
WaveFileNAME), 'file')
1430                                    copyfile(app.WaveFileLOC, app.inputFolder);
1431                                end
1432
1433                            end
1434
1435                        case "WK_DATA2D"
1436                            %----------------------
1437                            %......WK_DATA2D......
1438                            %----------------------
1439                            tempWavemaker2 = sprintf('WAVEMAKER = WK_DATA2D\n↙
WaveCompFile = Input_Data/%s\n Xc_WK = %.2f\n Yc_WK = %.2f\nDEP_WK = %.2f\nDelta_WK↙
= %.2f\n ', app.WavecomponentfileEditField.Value,Xc,Yc,DEP,app.WidthDeltaEditField.↙
Value);
1440
1441                            %Log report
1442                            checkAndReport(app.XcoordinatemEditField, 'X↙
Coordinate', fileId);
1443                            checkAndReport(app.YcoordinatemEditField, 'Y↙
Coordinate', fileId);
1444                            checkAndReport(app.WidthDeltaEditField, 'Width Delta',↙
fileId);
1445                            checkAndReport(app.WaterdepthmEditField, 'Water depth↙
(m)', fileId);
1446
```

```
1447                            %Check if the file is loaded
1448                            wavecompconditions = [...
1449                                isempty(app.WavecomponentfileEditField.Value ), ...
1450                                ~any(strcmp(app.WavecomponentfileEditField.Value , app.WaveFileNAME)), ...
1451                                str2double(app.WavecomponentfileEditField.Value ) == 0 ...
1452                                ];
1453
1454                            if any(wavecompconditions)
1455                                %Display 'NO FILE' in the textbox and note the issue in the log report
1456                                fprintf(fileId,'%-30s%-s\n','INITIAL CONDITION:','Wavemaker:');
1457                                fprintf(fileId, '%-35s%-s\n', '', 'No file uploaded in ''Wave component file''');
1458                                app.WavecomponentfileEditField.Value  = "NO FILE";
1459                                app.WavecomponentfileEditField.FontColor = 'w';
1460                                app.WavecomponentfileEditField.BackgroundColor = 'r';
1461                                error = error + 1;
1462                            elseif strcmp(app.WavecomponentfileEditField.Value , app.WaveFileNAME)
1463                                % Check if the uploaded file is not in the input folder
1464                                if ~exist(fullfile(app.inputFolder, app.WaveFileNAME), 'file')
1465                                    % If the file is not in 'INPUT_FILES/Input_Data', copy it there
1466                                    copyfile(app.WaveFileLOC, app.inputFolder);
1467                                end
1468                            end
1469
1470                        case "INI_GAUSSIAN"
1471                            %------------------------
1472                            %......INI_GAUSSIAN......
1473                            %------------------------
1474                            tempWavemaker2 = sprintf('WAVEMAKER = INI_GAUSSIAN\nAMP = %.2f\nXc = %.4f\nYc = %.4f\nWID = %.1f\n', AMP,Xc, Yc, WID);
1475
1476                            %Log report
1477                            checkAndReport(app.AmplitudemEditField, 'Amplitude (m)', fileId);
1478                            checkAndReport(app.XcoordinatemEditField, 'X Coordinate', fileId);
1479                            checkAndReport(app.YcoordinatemEditField, 'Y Coordinate', fileId);
1480                            checkAndReport(app.WidthmEditField, 'Width (m)', fileId);
1481
1482                        case "See options"
1483                            fprintf(fileId,'%-30s%-s\n','INITIAL CONDITION:','No Wavemaker type selected');
1484                            tempWavemaker2 = sprintf('!**No Wavemaker type selected\n');
1485                            error = error + 1;
1486                    end
```

```
1487
1488                %Combine all lines
1489                tempWavemaker = [tempWavemaker1, tempWavemaker2];
1490
1491                %Add these lines in the input.txt
1492                fprintf(fileInputId, '%s', tempWavemaker);
1493            end
1494
1495            %-----------------------------------------
1496            %------INITIAL CONDITION: Meteotsunami------
1497            %-----------------------------------------
1498            if app.MeteotsunamiButton.Value
1499                tempMeteo = sprintf('\n! ----------------↙
METEOTSUNAMI----------------\n');
1500
1501                %------------------------
1502                %......Air Pressure......
1503                %------------------------
1504                if strcmp(app.AirpressureSwitch.Value, "On")
1505                    tempMeteo = [tempMeteo, sprintf('MeteoGausian = T\n')];
1506
1507                   %Check if the pressure file exists
1508                    Pressureconditions = [...
1509                        isempty(app.PressurefileEditField.Value), ... % Check↙
if the field is empty
1510                        ~any(strcmp(app.PressurefileEditField.Value,  app.↙
PressureFileNAME)), ... % Check if the value does not match expected names
1511                        str2double(app.PressurefileEditField.Value) == 0 ... %↙
Check if the value is zero (assuming numeric comparison is relevant)
1512                        ];
1513
1514                    if any(Pressureconditions)
1515                        %Display 'NO FILE' in the textbox and note the issue↙
in the log report
1516                        app.PressurefileEditField.BackgroundColor = 'r';
1517                        app.PressurefileEditField.Value = "NO FILE";
1518                        app.PressurefileEditField.FontColor = 'w';
1519                        tempMeteo = [tempMeteo, sprintf('METEO_GAUSIAN_FILE =↙
Input_Data/\n')];
1520                        fprintf(fileId,'%-30s%-s\n','INITIAL↙
CONDITION:','Meteotsunami');
1521                        fprintf(fileId,'%-35s%-s\n','','No file uploaded in↙
''Pressure file''');
1522                        error = error + 1;
1523                        error_mkfile = error_mkfile + 1;
1524                        error_mkfiletext9 = sprintf('INITIAL CONDITION:↙
Meteotsunami');
1525                    else
1526                        tempMeteo = [tempMeteo, sprintf('METEO_GAUSIAN_FILE =↙
Input_Data/%s\n', string(app.PressureFileNAME))];
1527                        %If the file is not in 'INPUT_FILES/Input_Data', copy↙
it there
1528                        if ~exist(fullfile(app.inputFolder, app.↙
PressureFileNAME), 'file')
1529                            copyfile(app.PressureFileLOC, app.inputFolder);
1530                        end
1531                    end
1532                else
```

```
1533                        %Set the value to False
1534                        tempMeteo = [tempMeteo, sprintf('MeteoGaussian = F\n')];
1535                    end
1536
1537                    %-------------------------------
1538                    %......Constant Wind Field......
1539                    %-------------------------------
1540                    if strcmp(app.ConstantwindfieldSwitch.Value, "On")
1541                        tempMeteo = [tempMeteo, sprintf('WindConstantField =↙
T\n')];
1542
1543                        %Check if the uploaded Constant wind file exists
1544                        WindWaveconditions = [...
1545                            isempty( app.ConstantwindfileEditField.Value), ... %↙
Check if the field is empty
1546                            ~any(strcmp( app.ConstantwindfileEditField.Value,↙
app.ConsWindNAME)), ... % Check if the value does not match expected names
1547                            str2double( app.ConstantwindfileEditField.Value) == 0↙
... % Check if the value is zero (assuming numeric comparison is relevant)
1548                            ];
1549
1550                        if any(WindWaveconditions)
1551                            tempMeteo = [tempMeteo, sprintf('CONSTANT_WIND_FILE =↙
Input_Data/\n')];
1552
1553                            %Display 'NO FILE' in the textbox and note the issue↙
in the log report
1554                            app.ConstantwindfileEditField.BackgroundColor = 'r';
1555                            app.ConstantwindfileEditField.Value = "NO FILE";
1556                            app.ConstantwindfileEditField.FontColor = 'w';
1557
1558                            fprintf(fileId,'%-30s%-s\n','INITIAL↙
CONDITION:','Meteotsunami');
1559                            fprintf(fileId,'%-35s%-s\n','','No file uploaded in↙
''Pressure file''');
1560                            error = error + 1;
1561                            error_mkfile = error_mkfile + 1;
1562                            error_mkfiletext9 = sprintf('INITIAL CONDITION:↙
Meteotsunami');
1563                        else
1564                            tempMeteo = [tempMeteo, sprintf('CONSTANT_WIND_FILE =↙
Input_Data/%s\n', string(app.ConsWindNAME))];
1565                            %If the file is not in 'INPUT_FILES/Input_Data', copy↙
it there
1566                            if ~exist(fullfile(app.inputFolder, app.ConsWindNAME),↙
'file')
1567                                copyfile(app.ConsWindLOC, app.inputFolder);
1568                            end
1569                        end
1570                    else
1571                        %Set the value to False
1572                        tempMeteo = [tempMeteo, sprintf('WindConstantField =↙
F\n')];
1573                    end
1574
1575                    %---------------------------------
1576                    %......Wind Wave Interaction......
1577                    %---------------------------------
```

```
1578                 if strcmp(app.WindforceSwitch.Value, "On")
1579                     tempMeteo = [tempMeteo, sprintf('WindForce = T\nCdw = %.↙
3g\n', app.WindstresscoefficientEditField.Value)];
1580                     app.WindstresscoefficientEditField.FontColor = 'k';
1581 
1582                     % Wind Wave Interaction settings
1583                     if strcmp(app.WindwaveinteractionSwitch.Value, "On")
1584                         tempMeteo = [tempMeteo, sprintf('WindWaveInteraction =↙
T\nWindCrestPercent = %.2f\n', app.CrestRatioEditField.Value)];
1585                         app.CrestRatioEditField.FontColor = 'k';
1586                     else
1587                         %Set the value to False
1588                         tempMeteo = [tempMeteo, sprintf('WindWaveInteraction =↙
F\n')];
1589                     end
1590                 else
1591                     %Set the value to False
1592                     tempMeteo = [tempMeteo, sprintf('WindForce = F\n')];
1593                 end
1594 
1595                 %-------------------------------
1596                 %......Holland Storm Model......
1597                 %-------------------------------
1598                 if strcmp(app.HollandstormmodelSwitch.Value, "On")
1599                     tempMeteo = [tempMeteo, sprintf('WindHollandModel =↙
T\n')];
1600 
1601                     %Check if the uploaded Storm file exists
1602                     Stormconditions = [...
1603                         isempty(app.StormfileEditField.Value), ... % Check if↙
the field is empty
1604                         ~any(strcmp(app.StormfileEditField.Value,app.↙
StormFileNAME)), ... % Check if the value does not match expected names
1605                         str2double(app.StormfileEditField.Value) == 0  ... %↙
Check if the value is zero (assuming numeric comparison is relevant)
1606                         ];
1607 
1608                     if any(Stormconditions)
1609                         tempMeteo = [tempMeteo, sprintf('STORM_FILE =↙
Input_Data/\n')];
1610 
1611                         %Display 'NO FILE' in the textbox and note the issue↙
in the log report
1612                         app.StormfileEditField.BackgroundColor = 'r';
1613                         app.StormfileEditField.Value = "NO FILE";
1614                         app.StormfileEditField.FontColor = 'w';
1615 
1616                         fprintf(fileId,'%-30s%-s\n','INITIAL↙
CONDITION:','Meteotsunami');
1617                         fprintf(fileId,'%-35s%-s\n','','No file uploaded in↙
''Storm file''');
1618                         error = error + 1;
1619                         error_mkfile = error_mkfile + 1;
1620                         error_mkfiletext9 = sprintf('INITIAL CONDITION:↙
Meteotsunami');
1621                     else
1622                         tempMeteo = [tempMeteo, sprintf('STORM_FILE =↙
Input_Data/%s\n', string(app.StormFileNAME))];
```

```
1623                        % Check if the uploaded file is not from the INPUT_FOLDER
1624                        if ~exist(fullfile(inputFolder, app.StormFileNAME), 'file')
1625                            copyfile(app.StormFileLOC, inputFolder);
1626                        end
1627                    end
1628                else
1629                    %Set parameter to False
1630                    tempMeteo = [tempMeteo, sprintf('WindHollandModel = F\n')];
1631                end
1632
1633                %Add these lines in the input.txt
1634                fprintf(fileInputId, '%s', tempMeteo);
1635            end
1636
1637
1638
1639            %-----------------------------------------------------
1640            %------INITIAL CONDITION: Sediment Transport------
1641            %-----------------------------------------------------
1642            tempSediment = sprintf('\n! ---------------- SEDIMENT TRANSPORT ---------------- \n');
1643
1644            if app.SedimentTransportCheckBox.Value
1645                %--------------------
1646                %......Sediment......
1647                %--------------------
1648                if app.TVDButton.Value
1649                    tempSediment = [tempSediment, sprintf('Sed_Scheme = TVD \n')];
1650                elseif app.UpwindingButton.Value
1651                    tempSediment = [tempSediment, sprintf('Sed_Scheme = Upwinding \n')];
1652                end
1653
1654                editFields = [app.TimeStepSedToMorphlEditField, app.MinDepthSedPickUpEditField, ...
1655                    app.RungeKuttaparameter2EditField, app.RungeKuttaparameter1EditField, ...
1656                    app.ShieldsparamBedloadEditField, app.CriticalShieldsEditField, ...
1657                    app.SettlingvelocitymsEditField, app.SedimentporosityEditField, ...
1658                    app.SedimentdensityEditField, app.MediangrainDMEditField];
1659
1660                for field1 = editFields
1661                    field1.FontColor = 'k';
1662                end
1663
1664                %Median Grain diameter
1665                tempSediment = [tempSediment, sprintf('D50 = %.3f \n', app.MediangrainDMEditField.Value)];
1666
1667                %Sediment density
1668                tempSediment = [tempSediment, sprintf('Sdensity = %.3f \n', app.SedimentdensityEditField.Value)];
```

```
1669
1670                    %Sediment porosity
1671                    tempSediment = [tempSediment, sprintf('n_porosity = %.3f \n',↙
app.SedimentporosityEditField.Value)];
1672
1673                    %Settling velocity
1674                    tempSediment = [tempSediment, sprintf('WS = %.3f \n', app.↙
SettlingvelocitymsEditField.Value)];
1675
1676                    %Shields parameters (suspended load)
1677                    tempSediment = [tempSediment, sprintf('Shields_cr = %.3f \n',↙
app.CriticalShieldsEditField.Value)];
1678
1679                    %Shields parameters (bedload)
1680                    tempSediment = [tempSediment, sprintf('Shields_cr_bedload = %.↙
3f \n', app.ShieldsparamBedloadEditField.Value)];
1681
1682                    %Runge-Kutta pameters 1 and2
1683                    tempSediment = [tempSediment, sprintf('Kappa1 = %.4f \n', app.↙
RungeKuttaparameter1EditField.Value)];
1684
1685                    tempSediment = [tempSediment, sprintf('Kappa2 = %.3f \n', app.↙
RungeKuttaparameter2EditField.Value)];
1686
1687                    %Minimum depth for sediment pickup action
1688                    tempSediment = [tempSediment, sprintf('MinDepthPickup = %.3f↙
\n', app.MinDepthSedPickUpEditField.Value)];
1689
1690                    %---------------------------------
1691                    %......Morphological Changes......
1692                    %---------------------------------
1693                    if app.UpdatedepthButton.Value
1694                        tempSediment = [tempSediment, sprintf('Bed_Change = T↙
\nBedLoad = T\n')];
1695
1696                    elseif app.NobedchangeButton.Value
1697                        tempSediment = [tempSediment, sprintf('Bed_Change = F↙
\n')];
1698
1699                    elseif app.Button_36.Value
1700                        tempSediment = [tempSediment, sprintf('Bed_Change = \n')];
1701
1702                        %Log report
1703                        fprintf(fileId,'%-30s%-s\n','INITIAL↙
CONDITION:','Sediment:');
1704                        fprintf(fileId,'%-35s%-s\n','','No type of Bed Change is↙
selected');
1705                        app.UpdatedepthButton.FontColor = 'r';
1706                        app.NobedchangeButton.FontColor = 'r';
1707                        error = error + 1;
1708                    end
1709
1710                    if app.HardbednoerosionButton.Value
1711                        if  app.FileEditField.Value == string(app.HardbottomNAME)
1712                            tempSediment = [tempSediment, sprintf('Hard_bottom =↙
T\n')];
1713                            tempSediment = [tempSediment, sprintf↙
('Hard bottom file = Input Data/%s\n', string(app.HardbottomNAME))];
```

```
1714
1715                         if ~exist(fullfile(app.inputFolder, app.↙
HardbottomNAME), 'file')
1716                             %If the file is not in 'INPUT_FILES/Input_Data',↙
copy it there
1717                             copyfile(app.HardbottomLOC, app.inputFolder);
1718                         end
1719                     else
1720                         tempSediment = [tempSediment, sprintf↙
('Hard_bottom_file = Input_Data/\n')];
1721
1722                         %Display 'NO FILE' in the textbox and note the issue↙
in the log report
1723                         app.FileEditField.Value = "NO FILE";
1724                         app.FileEditField.FontColor = 'w';
1725                         app.FileEditField.BackgroundColor = 'r';
1726                         fprintf(fileId,'%-30s%-s\n','INITIAL↙
CONDITION:','Sediment:');
1727                         fprintf(fileId,'%-35s%-s\n','','No file uploaded in↙
''Hard Bottom File''');
1728
1729                         error = error + 1;
1730                         error_mkfile = error_mkfile + 1;
1731                         error_mkfiletext11 = sprintf('Initial Condition:↙
Sediment');
1732
1733                     end
1734                 elseif app.SedimentbedButton.Value
1735                     %Set value to False
1736                     tempSediment = [tempSediment, sprintf('Hard_bottom =↙
F\n')];
1737                 elseif app.Button_37.Value
1738                     tempSediment = [tempSediment, sprintf('Hard_bottom =↙
\n')];
1739
1740                     %Note the issue in the log report
1741                     fprintf(fileId,'%-30s%-s\n','INITIAL↙
CONDITION:','Sediment:');
1742                     fprintf(fileId,'%-35s%-s\n','','No Bed Type is selected');
1743                     app.HardbednoerosionButton.FontColor = 'r';
1744                     app.SedimentbedButton.FontColor = 'r';
1745                     error = error + 1;
1746                 end
1747
1748                 %---------------------
1749                 %......Avalanche......
1750                 %---------------------
1751                 if app.YesButton_8.Value
1752                     app.RungeKuttaparam2EditField_2.FontColor = 'k';
1753                     tempSediment = [tempSediment, sprintf('Avalanche = T↙
\n')];
1754                     tempSediment = [tempSediment, sprintf('Tan_phi = %.3g \n',↙
app.RungeKuttaparam2EditField_2.Value)];
1755                 end
1756
1757                 %Add these lines in the input.txt
1758                 fprintf(fileInputId, '%s', tempSediment);
1759
```

```
1760            end
1761
1762
1763            %-------------------------------------
1764            %------INITIAL CONDITION: Vessel------
1765            %-------------------------------------
1766            if app.VesselButton.Value
1767                %---------------------
1768                %......Vessel......
1769                %---------------------
1770                tempVessel1 = sprintf('\n! ---------------- SHIP WAKES↙
----------------------------\n');
1771
1772                %Check if the file is loaded
1773                Vesselconditions = [...
1774                    isempty(app.ImportvesselfilesTextArea.Value), ...
1775                    ~any(strcmp( app.ImportvesselfilesTextArea.Value, app.↙
VesselNAME)), ...
1776                    str2double( app.ImportvesselfilesTextArea.Value) == 0 ...
1777                    ];
1778
1779                if any(Vesselconditions)
1780                    tempVessel2 = sprintf(['VESSEL_FOLDER = ./\n' ...
1781                        'NumVessel = \n']);
1782                    %Display 'NO FILE' in the textbox and note the issue in↙
the log report
1783                    fprintf(fileId, '%-30s%-s\n', 'INITIAL CONDITION:',↙
'Vessel:');
1784                    fprintf(fileId, '%-35s%-s\n', '', 'No file uploaded in↙
''Import vessel file/s:''');
1785                    app.ImportvesselfilesTextArea.Value = "NO FILE";
1786                    app.ImportvesselfilesTextArea.FontColor = 'w';
1787                    app.ImportvesselfilesTextArea.BackgroundColor = 'r';
1788
1789                    error = error + 1;
1790                    error_mkfile = error_mkfile + 1;
1791                    error_mkfiletext8 = sprintf('Initial Condition: Vessel');
1792                else
1793                    tempVessel2 = sprintf(['VESSEL_FOLDER = %s\n' ...
1794                        'NumVessel = %s\n'], fullfile('.', 'Input_Data/'),↙
num2str(app.VesselNumber));
1795
1796                    for g = 1:length(app.ImportvesselfilesTextArea.Value)
1797                     %Copy files to'INPUT_FILES/Input_Data'
1798                    copyfile(app.VesselLOC(g), app.inputFolder);
1799                    end
1800                end
1801
1802                %Combine all the lines
1803                tempVessel = [tempVessel1,tempVessel2];
1804
1805                if app.ActivateCheckBox.Value
1806                    %------------------------------
1807                    %......Deep Draft Vessels......
1808                    %------------------------------
1809                    %Options for Methods:
1810
1811                    %Friction
```

```
1812                    if app.FrictionCheckBox_2.Value
1813                        frictionCoefficient = app.↙
FrictionCoefficientEditField.Value;
1814                        tempVessel = [tempVessel, sprintf('FrictionMethod =↙
T\nCdDeepDraft = %.1f\n', frictionCoefficient)];
1815                        app.FrictionCoefficientEditField.FontColor = 'k';
1816                    else
1817                        tempVessel = [tempVessel, sprintf('FrictionMethod =↙
F\n')];
1818                    end
1819
1820                    %Viscosity
1821                    if app.ViscosityCheckBox.Value
1822                        viscosityValue = app.ValueEditField.Value;
1823                        tempVessel = [tempVessel, sprintf('ViscosityMethod =↙
T\nVisDeepDraft = %.1f\n', viscosityValue)];
1824                        app.ValueEditField.FontColor = 'k';
1825                    else
1826                        tempVessel = [tempVessel, sprintf('ViscosityMethod =↙
F\n')];
1827                    end
1828
1829                    %Shock capturing
1830                    if app.ShockcapturingCheckBox.Value
1831                        tempVessel = [tempVessel, sprintf('MaskMethod =↙
T\n')];
1832                    else
1833                        tempVessel = [tempVessel, sprintf('MaskMethod =↙
F\n')];
1834                    end
1835
1836                    minimumClearance = app.MinimumClearanceEditField.Value;
1837
1838                    %Add another line at the end of tempVessel
1839                    tempVessel = [tempVessel, sprintf('CLEARANCE = %.1f\n',↙
minimumClearance)];
1840                    app.MinimumClearanceEditField.FontColor = 'k';
1841                end
1842
1843                %Add these lines in the input.txt
1844                fprintf(fileInputId, '%s', tempVessel);
1845            end
1846
1847            %----------------------------------------
1848            %------PERIODIC BOUNDARY CONDITION------
1849            %----------------------------------------
1850            tempPBC = sprintf([
1851                '\n ! ---------------- PERIODIC BOUNDARY CONDITION ---------↙
\n' ...
1852                '   ! South-North periodic boundary condition\n' ...
1853                'PERIODIC = F\n']);
1854
1855            %Add these lines in the input.txt
1856            fprintf(fileInputId, '%s',tempPBC);
1857
1858            %---------------------------
1859            %------SPONGE BOUNDARY------
1860            %---------------------------
```

```
1861            tempSponge = sprintf([
1862                '\n  ! --------------- SPONGE LAYER ------------------------↙
\n' ...
1863                '  ! DHI type sponge layer\n' ...
1864                '  ! need to specify widths of four boundaries and↙
parameters\n' ...
1865                '  ! set width=0.0 if no sponge\n' ...
1866                '  ! R_sponge: decay rate\n' ...
1867                '  ! A_sponge: maximum decay rate\n' ...
1868                '  ! e.g., sharp: R=0.85 \n' ...
1869                '  !       mild:  R=0.90, A=5.0 \n' ...
1870                '  !       very mild, R=0.95, A=5.0 \n']);
1871
1872
1873            %Add these lines in the input.txt
1874            fprintf(fileInputId, '%s',tempSponge);
1875
1876            % Set sponge dimensions if any sponge setting is active
1877            if app.DirectCheckBox.Value || app.FrictionCheckBox.Value || app.↙
DiffusionCheckBox.Value
1878                spongeOn = sprintf('SPONGE_ON = T \n');
1879            else
1880                spongeOn = sprintf('SPONGE_ON = F \n');
1881            end
1882            fprintf(fileInputId, '%s',spongeOn);
1883
1884            %Direct Sponge
1885            if app.DirectCheckBox.Value
1886                fprintf(fileInputId, 'DIRECT_SPONGE = T \n');
1887
1888                % Determine the decay rate and associated sponge settings
1889                switch app.DecayTypeDropDown.Value
1890                    case 'Very mild'
1891                        decayRate = sprintf([ ...
1892                            'R_sponge = 0.95 \n' ...
1893                            'A_sponge = 5.0 \n']);
1894                    case 'Mild'
1895                        decayRate = sprintf([ ...
1896                            'R_sponge = 0.9 \n' ...
1897                            'A_sponge = 5.0 \n']);
1898                    case 'Sharp'
1899                        decayRate = sprintf('R_sponge = 0.85 \n');
1900                end
1901
1902                %Add these lines in the input.txt
1903                fprintf(fileInputId, '%s', decayRate);
1904            end
1905
1906            %Friction
1907            spongeSetting = '';
1908            if app.FrictionCheckBox.Value
1909                spongeSetting = sprintf('FRICTION_SPONGE = T\nCDsponge = %.3f↙
\n', app.MaximumCdEditField.Value);
1910                %Add these lines in the input.txt
1911                fprintf(fileInputId, '%s', spongeSetting);
1912            end
1913
1914            %Diffusion
```

```
1915            if app.DiffusionCheckBox.Value
1916                spongeSetting = sprintf('DIFFUSION_SPONGE = T\nCsp = %.3f \n',↙
app.MaximumCspEditField.Value);
1917                %Add these lines in the input.txt
1918                fprintf(fileInputId, '%s', spongeSetting);
1919            end
1920
1921            %Add these lines in the input.txt
1922            % fprintf(fileInputId, '%s', spongeSetting);
1923
1924
1925            % Set sponge dimensions if any sponge setting is active
1926            if app.DirectCheckBox.Value || app.FrictionCheckBox.Value || app.↙
DiffusionCheckBox.Value
1927                spongeBound = sprintf([
1928                    'Sponge_west_width = %.1f\n' ...
1929                    'Sponge_east_width = %.1f\n' ...
1930                    'Sponge_south_width = %.1f\n' ...
1931                    'Sponge_north_width = %.1f\n'], ...
1932                    app.WestEditField.Value, app.EastEditField.Value, app.↙
SouthEditField.Value, app.NorthEditField.Value);
1933
1934                %Log warnings when the sponge layer is turned ON but the↙
boundaries are set to zero
1935                if app.NorthEditField.Value == 0
1936                    app.NorthEditField.FontColor = 'r';
1937                    fprintf(fileId,'%-30s%-s\n','SPONGE BOUNDARY:', 'North↙
buffer is zero');
1938                end
1939                if app.EastEditField.Value == 0
1940                    app.EastEditField.FontColor = 'r';
1941                    fprintf(fileId,'%-30s%-s\n','SPONGE BOUNDARY:', 'East↙
buffer is zero');
1942                end
1943                if app.WestEditField.Value == 0
1944                    app.WestEditField.FontColor = 'r';
1945                    fprintf(fileId,'%-30s%-s\n','SPONGE BOUNDARY:', 'West↙
buffer is zero');
1946                end
1947                if app.SouthEditField.Value == 0
1948                    app.SouthEditField.FontColor = 'r';
1949                    fprintf(fileId,'%-30s%-s\n','SPONGE BOUNDARY:', 'South↙
buffer is zero');
1950                end
1951
1952            else
1953                spongeBound = sprintf([
1954                    'Sponge_west_width = 0.0\n' ...
1955                    'Sponge_east_width = 0.0\n' ...
1956                    'Sponge_south_width = 0.0\n' ...
1957                    'Sponge_north_width = 0.0\n']);
1958
1959                app.NorthEditField.FontColor = '0.65,0.65,0.65';
1960                app.EastEditField.FontColor = '0.65,0.65,0.65';
1961                app.WestEditField.FontColor = '0.65,0.65,0.65';
1962                app.SouthEditField.FontColor = '0.65,0.65,0.65';
1963            end
1964
```

```
1965            %Add these lines in the input.txt
1966            fprintf(fileInputId, '%s', spongeBound);
1967
1968            %---------------------
1969            %------OBSTACLES------
1970            %---------------------
1971            if app.AddobstaclesCheckBox.Value
1972                if isempty(app.ObstacleFilename.Value) || ~strcmp(app.↙
ObstacleFilename.Value, app.obstacleNAME) || ~exist(app.obstacleLOC, "file")
1973                    % Display 'NO FILE' in the textbox
1974                    app.ObstacleFilename.Value = "NO FILE";
1975                    app.ObstacleFilename.FontColor = 'w';
1976                    app.ObstacleFilename.BackgroundColor = 'r';
1977                    fprintf(fileId, '%-30s%-s\n', 'OBSTACLES:', 'No file is↙
uploaded');
1978                    error = error + 1;
1979                else
1980                    if strcmp(app.ObstacleFilename.Value, string(app.↙
obstacleNAME))
1981                        tempObstacle = sprintf([
1982                            '  \n! ----------------↙
OBSTACLES----------------------------\n' ...
1983                            '  ! obstacle structures using mask_struc file\n'↙
...
1984                            '  ! mask_struc =0 means structure element\n' ...
1985                            '  ! give a file contains a mask array with Mloc X↙
Nloc\n' ...
1986                            '  OBSTACLE_FILE= %s\n'], ...
1987                            app.ObstacleFilename.Value);
1988
1989                        % Add these lines in the input.txt
1990                        fprintf(fileInputId, '%s', tempObstacle);
1991
1992                        if exist(app.obstacleLOC, "file")
1993                            obstacle = readmatrix(app.obstacleLOC);
1994                            if isnumeric(obstacle)
1995                                if ~all(obstacle(:) == 0 | obstacle(:) == 1)
1996                                    fprintf(fileId, '%-30s%-s\n',↙
'OBSTACLES:', 'File should only contain values of 1 and 0');
1997                                end
1998                            end
1999                        else
2000                            fprintf(fileId, '%-30s%-s\n', 'OBSTACLES:',↙
'Uploaded file is empty');
2001                        end
2002
2003                        % If the file is not in 'INPUT_FILES/Input_Data', copy↙
it there
2004                        if ~exist(fullfile(inputFolder, app.obstacleNAME),↙
'file')
2005                            copyfile(app.obstacleLOC, inputFolder);
2006                        end
2007                    end
2008                end
2009            end
2010
2011            %-------------------
2012            %------PHYSICS------
```

```
2013            %-------------------
2014            tempPhysics1 = sprintf([
2015                '\n  ! ----------------PHYSICS-------------------------------↙
\n' ...
2016                '  ! parameters to control type of equations\n' ...
2017                '  ! dispersion: all dispersive terms\n' ...
2018                '  ! gamma1=1.0,gamma2=0.0: NGs equations\n' ...
2019                '  ! gamma1=1.0,gamma2=1.0: Fully nonlinear equations\n']);
2020
2021            %Initial set up for the second part of the lines for this section
2022            tempPhysics2 = '';
2023
2024            %-----------------------------------------
2025            %......Linear Shallow Water Equation......
2026            %-----------------------------------------
2027            if app.LinearShallowWaterEquationButton.Value
2028                app.LinearShallowWaterEquationButton.FontColor = 'k';
2029                tempPhysics2 = sprintf([
2030                    'Gamma1 = 0.0\n' ...
2031                    'Gamma2 = 0.0\n' ...
2032                    'Gamma3 = 0.0\n']);
2033
2034                %-----------------------------------------
2035                %......Nonlinear Shallow Water Equation......
2036                %-----------------------------------------
2037            elseif app.NonlinearShallowWaterEquationButton.Value
2038                app.NonlinearShallowWaterEquationButton.FontColor = 'k';
2039                tempPhysics2 = sprintf([
2040                    'Gamma1 = 0.0\n' ...
2041                    'Gamma2 = 0.0\n' ...
2042                    'Gamma3 = 1.0\n' ...
2043                    'Beta_ref=-0.531\n']);
2044
2045                %--------------------------------------
2046                %......Nonlinear Boussinesq Equation......
2047                %--------------------------------------
2048            elseif app.FullyNonlinearBoussinesqEquationButton.Value
2049                app.FullyNonlinearBoussinesqEquationButton.FontColor = 'k';
2050                tempPhysics2 = sprintf([
2051                    'Gamma1 = 1.0\n' ...
2052                    'Gamma2 = 1.0\n' ...
2053                    'Gamma3 = 1.0\n' ...
2054                    'Beta_ref=-0.531\n' ...
2055                    'SWE_ETA_DEP = 0.8\n'...
2056                    'DISPERSION = T\n']);
2057            end
2058
2059
2060
2061            if app.LinearShallowWaterEquationButton.Value || app.↙
FullyNonlinearBoussinesqEquationButton.Value || app.↙
NonlinearShallowWaterEquationButton.Value
2062                %Combine all the lines
2063                tempPhysics = [tempPhysics1 tempPhysics2];
2064
2065                %Add these lines in the input.txt
2066                fprintf(fileInputId, '%s', tempPhysics);
2067            elseif app.Button.Value
```

```
2068                %Display 'NO FILE' in the textbox and note the issue in the log report
2069                app.FullyNonlinearBoussinesqEquationButton.FontColor = 'r';
2070                app.LinearShallowWaterEquationButton.FontColor = 'r';
2071                app.NonlinearShallowWaterEquationButton.FontColor = 'r';
2072                fprintf(fileId,'%-30s%-s\n','PHYSICS:', 'No equation is selected');
2073                error = error + 1;
2074            end
2075
2076            %---------------------------
2077            %------BOTTOM FRICTION------
2078            %---------------------------
2079            % Initialize base friction information
2080            tempFriction1 = sprintf(' \n !----------------Friction-----------------------------\n');
2081
2082            tempFriction2 = '';  %Initial set up for the second part of the lines for this section
2083
2084            if app.ConstantCoefficientButton.Value
2085                %-----------------------------
2086                %......Constant Friction......
2087                %-----------------------------
2088                tempFriction2 = sprintf([
2089                    'Friction_Matrix= F\n' ...
2090                    'Cd = %.3f\n' ...
2091                    'Cd_fixed = %.3f\n'], ...
2092                    app.ConstantFrictionValue.Value, ...
2093                    app.ConstantFrictionValue.Value);
2094
2095                if app.ConstantFrictionValue.Value == 0
2096                    tempFriction2 = sprintf('Friction_Matrix= F \n  Cd = 0.0');
2097                end
2098
2099            elseif app.NonconstantButton.Value
2100                %-------------------------------
2101                %......Nonconstan Friction......
2102                %-------------------------------
2103                if isempty(app.FrictionFileEditField.Value) || ~strcmp(app.FrictionFileEditField.Value, app.filenamefriction) || ~exist(app.Cd_file,"file")
2104                    %Display 'NO FILE' in the textbox and note the issue in the log report
2105                    app.FrictionFileEditField.BackgroundColor = 'r';
2106                    app.FrictionFileEditField.Value = "NO FILE";
2107                    app.FrictionFileEditField.FontColor = 'w';
2108                    fprintf(fileId, '%-30s%-s\n', 'BOTTOM FRICTION:', 'No file is uploaded');
2109
2110                    error = error + 1;
2111                    error_mkfile = error_mkfile + 1;
2112                    error_mkfiletext7 = sprintf('Friction');
2113                else
2114                    tempFriction2 = sprintf('Friction_Matrix= T\nCd_file=Input_Data/%s\n', app.FrictionFileEditField.Value);
2115
2116                    if ~exist(fullfile(app.inputFolder, app.Cd_file), 'file')
```

```
2117                        % If the file is not in 'INPUT_FILES/Input_Data', copy it there
2118                        copyfile(app.Cd_file, app.inputFolder);
2119                    end
2120                end
2121            end
2122
2123            % Combine the parts
2124            tempFriction = [tempFriction1 tempFriction2];
2125
2126            %Add these lines in the input.txt
2127            fprintf(fileInputId, '%s', tempFriction);
2128
2129            %--------------------
2130            %------NUMERICS------
2131            %--------------------
2132            if app.LinearShallowWaterEquationButton.Value || app.NonlinearShallowWaterEquationButton.Value
2133                tempNumerics =  sprintf([
2134                    '\n  ! ----------------NUMERICS----------------------------\n'...
2135                    'HIGH_ORDER = SECOND \n' ...
2136                    'CFL = 0.5 \n' ...
2137                    '  ! Froude Number Cap (to avoid jumping drop, set 10) \n' ...
2138                    'FroudeCap = %.2f \n'],app.FroudeCapEditField.Value);
2139            else
2140                tempNumerics =  sprintf([
2141                    '\n  ! ----------------NUMERICS----------------------------\n'...
2142                    'HIGH_ORDER = THIRD \n' ...
2143                    'CFL = 0.5 \n' ...
2144                    '  ! Froude Number Cap (to avoid jumping drop, set 10) \n' ...
2145                    'FroudeCap = %.2f \n'],app.FroudeCapEditField.Value);
2146            end
2147
2148            %Add these lines in the input.txt
2149            fprintf(fileInputId, '%s', tempNumerics);
2150
2151            %-------------------------
2152            %------MINIMUM DEPTH------
2153            %-------------------------
2154            tempWetDry1 = sprintf('\n  ! --------------WET-DRY-----------------------------\n');
2155
2156            tempWetDry2 = sprintf([
2157                '  ! MinDepth for wetting-drying\n' ...
2158                'MinDepth= %.2f\n'], ...
2159                app.WetDrySchemeEditField.Value);
2160
2161            if app.ConstantCoefficientButton.Value || app.NonconstantButton.Value
2162                tempWetDry3 = sprintf([
2163                    '  ! MinDepthfrc to limit bottom friction\n' ...
2164                    'MinDepthFrc = %.2f \n'], ...
2165                    app.BottomFrictionEditField.Value);
2166
```

```
2167                 tempWetDry = [tempWetDry1, tempWetDry2, tempWetDry3];
2168             else
2169                 tempWetDry = [tempWetDry1, tempWetDry2];
2170             end
2171
2172             %Add these lines in the input.txt
2173             fprintf(fileInputId, '%s', tempWetDry);
2174
2175             %-------------------------
2176             %------WAVE BREAKING------
2177             %-------------------------
2178             if app.FullyNonlinearBoussinesqEquationButton.Value
2179                 tempShowBreaking1 = sprintf([
2180                     '\n  ! -------------- SHOW BREAKING↙
----------------------\n' ...
2181                     '  ! breaking is calculated using shock wave capturing↙
scheme\n' ...
2182                     '  ! the criteria is only for demonstration or bubble↙
calculation\n' ...
2183                     '  ! Cbrk1=0.45,Cbrk2=0.35, for irregular waves, there are↙
much small!\n']);
2184
2185                 if app.ShockwavecapturingButton.Value
2186                     tempShowBreaking2 = sprintf([
2187                         'SHOW_BREAKING = T\n' ...
2188                         'VISCOSITY_BREAKING = F\n']);
2189                 elseif app.ViscositybreakingButton.Value
2190                     tempShowBreaking2 = sprintf([
2191                         'SHOW_BREAKING = F\n' ...
2192                         'VISCOSITY_BREAKING = T\n'...
2193                         'Cbrk1 = %.2f\n' ...
2194                         'Cbrk2 = %.2f\n'], ...
2195                         app.C1EditField.Value, ...
2196                         app.C2EditField.Value);
2197                 else
2198                     tempShowBreaking2 = '';
2199                 end
2200
2201                 tempShowBreaking = [tempShowBreaking1, tempShowBreaking2];
2202                 fprintf(fileInputId, '%s', tempShowBreaking);
2203             end
2204
2205             if app.AddrollereffectsCheckBox.Value && ~app.↙
FullyNonlinearBoussinesqEquationButton.Value
2206                 tempShowBreaking1 = sprintf([
2207                     '\n  ! -------------- SHOW BREAKING↙
----------------------\n']);
2208                 tempShowBreaking2 = sprintf('ROLLER_EFFECT = T');
2209                 app.UndertowCheckBox.Value = 1; %Set the expected output↙
UNDERTOW = T
2210                 app.RollerinducedFluxCheckBox.Value = 1; %Set the expected↙
output Roller = T
2211
2212                 %Combine all lines
2213                 tempShowBreaking = [tempShowBreaking1, tempShowBreaking2];
2214
2215                 %Add these lines in the input.txt
2216                 fprintf(fileInputId, '%s\n', tempShowBreaking);
```

```
2217
2218            elseif app.AddrollereffectsCheckBox.Value && app.↙
FullyNonlinearBoussinesqEquationButton.Value
2219                fprintf(fileInputId, '%s\n','ROLLER_EFFECT = T');
2220            end
2221
2222            %------------------
2223            %------MIXING------
2224            %------------------
2225            if app.UseSmagorinskyCheckBox.Value
2226                tempMixing = sprintf([...
2227                    '\n  ! ----------------- MIXING↙
---------------------------\n' ...
2228                    '  ! if use smagorinsky mixing, have to set -DMIXING in↙
Makefile\n' ...
2229                    '  ! and set averaging time interval, T_INTV_mean,↙
default: 20s\n' ...
2230                    'T_INTV_mean = %.2f\n' ...
2231                    'STEADY_TIME = %.2f\n' ...
2232                    'C_smg = 0.25\n'], ...
2233                    app.TimeintervalsecEditField.Value, ...
2234                    app.SteadyTimeEditField.Value);
2235                fprintf(fileInputId, '%s', tempMixing);
2236            end
2237
2238            %----------------------------
2239            %------EXPECTED OUTPUT------
2240            %----------------------------
2241            tempOutput1 = sprintf('\n! -----------------↙
OUTPUT-----------------------------\n');
2242
2243            %------------------
2244            %------GAUGES------
2245            %------------------
2246            if app.ImportgaugelistCheckBox.Value
2247                %Use the loaded file as regular virtual gauges, not to be used↙
for nested grid
2248
2249                %Check if gauge file is loaded
2250                loadFileNum = str2double(app.LoadFileEditField.Value);
2251
2252                conditions = [...
2253                    isempty(app.LoadFileEditField.Value), ...
2254                    loadFileNum == 0, ...
2255                    ~any(strcmp(app.LoadFileEditField.Value,app.GAUGEFILE)) ↙
...
2256                    ];
2257
2258                if any(conditions)
2259                    tempOutput2 = sprintf('NumberStations = 0\n');
2260
2261                    %Display 'NO FILE' in the textbox and note the issue in↙
the log report
2262                    app.LoadFileEditField.BackgroundColor = 'r';
2263                    app.LoadFileEditField.Value = "NO FILE";
2264                    app.LoadFileEditField.FontColor = 'w';
2265                    fprintf(fileId,'%-30s%-s\n','GAUGES:', 'No file is↙
uploaded');
```

```
2266                        tempOutput = sprintf('NumberStations = 0\n');
2267                        error = error + 1;
2268
2269                        app.SpacingEditField.FontColor = '0.65,0.65,0.65';
2270                        app.WestEditField_2.FontColor = '0.65,0.65,0.65';
2271                        app.SouthEditField_2.FontColor = '0.65,0.65,0.65';
2272                        app.EastEditField_2.FontColor = '0.65,0.65,0.65';
2273                        app.NorthEditField_2.FontColor = '0.65,0.65,0.65';
2274
2275                    elseif app.LoadFileEditField.Value == string(app.GAUGEFILE)
2276                        tempOutput2 = sprintf([
2277                            'NumberStations = %s\n' ...
2278                            'STATIONS_FILE =  Input_Data/%s\n'], ...
2279                            app.NumberStations, ...
2280                            app.STATION_FILE);
2281
2282                        if contains(app.LoadFileEditField.Value, '.shp')
2283                            % If the file type is a shapefile, convert it to a↙
tab-delimited text file
2284                            outputTEXTFILE = fullfile(app.inputFolder, string(app.↙
GaugeFname) + '.txt');
2285                            writematrix(app.GaugeListGrids, outputTEXTFILE,↙
'Delimiter', '\t');
2286
2287                        elseif contains(app.LoadFileEditField.Value, '.txt')
2288                            % Check if all items are numeric
2289                            if isnumeric(app.gaugetext0)
2290                                % Log Report
2291                                if size(app.gaugetext0, 2) ~= 2
2292                                    fprintf(fileId, '%-30s%-s\n', 'GAUGES:', 'Make↙
sure that the .txt file contains 2 columns (Lat Long)');
2293                                    error = error + 1;
2294                                end
2295
2296                                if any(isnan(app.gaugetext0(:)))
2297                                    fprintf(fileId, '%-30s%-s\n', 'GAUGES:', 'Text↙
file contains NaN values');
2298                                    error = error + 1;
2299                                end
2300                            else
2301                                if ischar(app.gaugetext0) || isstring(app.↙
gaugetext0)
2302                                    if any(isletter(app.gaugetext0))
2303                                        fprintf(fileId, '%-30s%-s\n', 'GAUGES:',↙
'Uploaded .txt file contains non-numeric values');
2304                                        error = error + 1;
2305                                    end
2306                                end
2307                            end
2308
2309                            % If the file is not in 'INPUT_FILES/Input_Data', copy↙
it there
2310                            if ~exist(fullfile(inputFolder, app.GAUGEFILE),↙
'file')
2311                                copyfile(app.GAUGEFILElocation, inputFolder);
2312                            end
2313                        end
2314                    end
```

```
2315            else
2316                %Set value to zero
2317                tempOutput2 = sprintf('NumberStations = 0\n');
2318            end
2319
2320
2321
2322            if app.CreatenestedgridboundaryCheckBox.Value
2323                %Use the gauge stations as the boundary of the nested grid
2324
2325                %Create nested grid boundary code from Shi et al. (2012)
2326                % FUNWAVE version 3.6: simple cases > nesting tools > ↙
make_stations_single_block.m
2327
2328                dx= app.SpacingEditField.Value;
2329                id_count=0;
2330                icount=0;
2331
2332                x1 =  app.WestEditField_2.Value + (dx/2);
2333                x2 =  app.EastEditField_2.Value;
2334                y1 =  app.SouthEditField_2.Value + (dx/2);
2335                y2 =  app.NorthEditField_2.Value;
2336
2337
2338                % if app.CoupMatrixON == 0 %Use the coordinates of the ↙
uploaded .tif file
2339                %      %Assuming the coordinate of the SW corner of the tif ↙
file is measured at the pixel's corner.
2340                %      %This reads the coordinate at the centre of the pixel
2341                %     x1 = app.WestEditField_2.Value; %+ (dx/2);
2342                %     x2 = app.EastEditField_2.Value; %- (dx/2);
2343                %     y1 =  app.SouthEditField_2.Value; %+ (dx/2);
2344                %     y2 =  app.NorthEditField_2.Value; %- (dx/2);
2345                % end
2346
2347                %Create the list of coordinates
2348                x=[x1:dx:x2];
2349                y=[y1:dx:y2];
2350
2351
2352                XCouplingLength = length(x);
2353                YCouplingLength = length(y);
2354                id_count=id_count+1;
2355
2356                %Arrangement of E,W,S,N is based on the requirement of ↙
convert.f
2357                for j=1:length(y)
2358                    icount=icount+1;
2359                    app.NestStation(icount,1)=y(j);
2360                    app.NestStation(icount,2)=x(end);
2361                    app.NestStation(icount,3)=id_count;
2362                end
2363
2364                id_count=id_count+1;
2365                % west
2366                for j=1:length(y)
2367                    icount=icount+1;
2368                    app.NestStation(icount,1)=y(j);
```

```
2369                    app.NestStation(icount,2)=x(1);
2370                    app.NestStation(icount,3)=id_count;
2371                end
2372
2373
2374                id_count=id_count+1;
2375                % south
2376                for i=1:length(x)
2377                    icount=icount+1;
2378                    app.NestStation(icount,1)=y(1);
2379                    app.NestStation(icount,2)=x(i);
2380                    app.NestStation(icount,3)=id_count;
2381                end
2382
2383                id_count=id_count+1;
2384                % north
2385                for i=1:length(x)
2386                    icount=icount+1;
2387                    app.NestStation(icount,1)=y(end);
2388                    app.NestStation(icount,2)=x(i);
2389                    app.NestStation(icount,3)=id_count;
2390                end
2391
2392                fullFilePathNest = fullfile(app.inputFolder, 'nestingboundary.↙
txt');
2393                icount_last=icount;
2394
2395
2396                fid2 = fopen(fullFilePathNest, 'wt');
2397                fprintf(fid2, ['%f %f %d', '\n'], app.NestStation');
2398                fclose(fid2);
2399
2400
2401                tempOutput2 = sprintf([
2402                    'NumberStations = %d\n' ...
2403                    'STATIONS_FILE = Input_Data/nestingboundary.txt \n'], ...
2404                    icount_last);
2405
2406                roundup = ceil(icount_last / 500) * 500; %round up to the↙
nearest 500;
2407
2408
2409                %Setting up the info to be displayed in the log report
2410                formattedCouplingDetails = sprintf([
2411                    '%-30s%-s\n', ...
2412                    '%-30s%-s\n', ...
2413                    '%-30s%-s\n', ...
2414                    '%-30s%-s\n', ...
2415                    '%-30s%-s\n', ...
2416                    '%-30s%-s\n', ...
2417                    '%-30s%-s\n\n'], ...
2418                    'GAUGES:', '''nestingboundary.txt'' is created', ...
2419                    'COUPLING:', 'Details to input in convert.f for your next↙
nested grid run:', ...
2420                    '', ['nsta_tot and Nb:  ' num2str(roundup)], ...
2421                    '', ['nsta:  ' num2str(icount_last)], ...
2422                    ' ', ['neast and nwest: ' num2str(YCouplingLength)], ... %↙
num2str(app.CouplingX)], ...  %refers to the N matrix dimension
```

```
2423                    ' ', ['nnorth and nsouth: ' num2str(XCouplingLength)], ...
%, ... %num2str(app.CouplingY)])  %refers to the M matrix dimension
2424                    '','Ntotal refers to the total number of rows contained in
each output 'sta_' file generated after running your model',...
2425                    '','Update ''fdir'' to the directory path where your
'sta_' files and convert.f file are located',...
2426                    '','The ''convert.f'' file is located in
benchmarks/sph_nesting/make_nest_file');
2427
2428                %Formatting
2429                fprintf(fileId, formattedCouplingDetails);
2430
2431                %Log Report
2432                fields = {
2433                    app.SpacingEditField, 'Gauge spacing is zero';
2434                    app.WestEditField_2, 'West grid coordinate is zero';
2435                    app.EastEditField_2, 'East grid coordinate is zero';
2436                    app.SouthEditField_2, 'South grid coordinate is zero';
2437                    app.NorthEditField_2, 'North grid coordinate is zero'
2438                    };
2439
2440                %Loop through each field and check its value
2441                for i = 1:size(fields, 1)
2442                    field = fields{i, 1};
2443                    message = fields{i, 2};
2444
2445                    if field.Value == 0
2446                        field.FontColor = 'r';
2447                        fprintf(fileId, '%-30s%-s\n', 'GAUGES:', message);
2448                        error = error + 1;
2449                    end
2450                end
2451
2452
2453            end
2454
2455            %Combine all the lines and add them in the input.txt
2456            tempOutput = [tempOutput1,tempOutput2];
2457            fprintf(fileInputId, '%s', tempOutput);
2458
2459
2460            %----------------------------
2461            %------Expected OUTPUT------
2462            %----------------------------
2463            commands = struct( ...
2464                'DepthCheckBox', 'DEPTH_OUT = T \n', ...
2465                'UVelocityCheckBox', 'U = T \n', ...
2466                'VVelocityCheckBox', 'V = T \n', ...
2467                'UVelocitymeanCheckBox', 'Umean = T \n', ...
2468                'VVelocitymeanCheckBox', 'Vmean = T \n', ...
2469                'SurfaceElevationattimetCheckBox', 'ETA = T \n', ...
2470                'WetdrymaskCheckBox', 'MASK = T \n', ...
2471                'WetdrymaskforBoussinesqNSWECheckBox', 'MASK9 = T \n', ...
2472                'XSourceCheckBox', 'SourceX = T \n', ...
2473                'YSourceCheckBox', 'SourceY = T \n', ...
2474                'PMomentumFluxCheckBox', 'P = T \n', ...
2475                'QmomentumFluxCheckBox', 'Q = T \n', ...
2476                'FxFluxCheckBox', 'Fx = T \n', ...
```

```
2477                'GxFluxCheckBox', 'Gx = T \n', ...
2478                'GyFluxCheckBox', 'Gy = T \n', ...
2479                'BreakingAgeCheckBox', 'AGE = T \n', ...
2480                'MaximumWaveHeightCheckBox', 'Hmax = T \n', ...
2481                'FyFluxCheckBox', 'Fy = T \n', ...
2482                'MinimumWaveHeightCheckBox', 'Hmin = T \n', ...
2483                'MaximumVelocityCheckBox', 'Umax = T \n', ...
2484                'UndertowCheckBox', 'UNDERTOW = T \n', ...
2485                'MaxVorticityCheckBox', 'VORmax = T \n', ...
2486                'MaxMomentumFluxCheckBox', 'MFmax = T \n', ...
2487                'ArrivalCheckBox', 'OUT_Time = T \n', ...
2488                'WaveHeightCheckBox', 'WaveHeight = T \n', ...
2489                'RollerinducedFluxCheckBox', 'ROLLER = T \n', ...
2490                'PressureFieldCheckBox', 'OUT_METEO = T \n', ...
2491                'BreakingLocationCheckBox', 'OUT_NU = T \n' ...
2492                );
2493
2494
2495            %Loop through each item. If the values are checked in the app, list each variable and set them to 'T' in the input.txt
2496            fields = fieldnames(commands);
2497            for i = 1:length(fields)
2498                fieldName = fields{i};
2499                if app.(fieldName).Value
2500                    fprintf(fileInputId, commands.(fieldName));
2501                end
2502            end
2503
2504            %--------------------------
2505            %------GENERAL ERRORS------
2506            %--------------------------
2507            if error ~=0 %Errors are present
2508                %Updating the log report
2509                fprintf(fileId, '%s\n', '');
2510                fprintf(fileId, '%s\n','> Status: Creating input.txt... Completed WITH ISSUES');
2511                fprintf(fileId, '%s\n','> Resolve the warnings above to prevent modeling errors later');
2512                app.Preview_LogReport.Value = fileread(LogFile); %show the log report
2513                drawnow
2514
2515            else
2516                %Updating the log report
2517                fprintf(fileId, '%s\n', '');
2518                fprintf(fileId, '%s\n\n','> Status: Creating input.txt... Succesfully completed!');
2519                app.Preview_LogReport.Value = fileread(LogFile); %show the log report
2520                drawnow
2521            end
2522
2523            %--------------------------
2524            %------MAKEFILE ERRORS------
2525            %--------------------------
2526            %Checks if there are issues associated with Makefile creation
2527            if error_mkfile ~= 0
2528                %Updating the log report
```

```
2529                fprintf(fileId, '%s\n','> Status: The Makefile section will be enabled once the raised issues are resolved');
2530                fprintf(fileId, '%s\n','   Specifically, check the following sections:');
2531
2532                % Define a list of variables to check
2533                variablesToCheck = {'error_mkfiletext3', 'error_mkfiletext5', 'error_mkfiletext7', 'error_mkfiletext8', 'error_mkfiletext9', 'error_mkfiletext11'};
2534                flags = {'', '', '', '', 'flag10', ''};  % Associated flags for each variable
2535
2536                % Loop through each variable and check if it exists
2537                for i = 1:length(variablesToCheck)
2538                    variableName = variablesToCheck{i};
2539                    flag = flags{i};
2540
2541                    % Use eval to check if variable exists and is not empty
2542                    if eval(sprintf('exist(''%s'', ''var'') && ~isempty(%s)', variableName, variableName))
2543                        % Print the variable's value
2544                        fprintf(fileId, '\t%2s\n', eval(variableName));
2545
2546                        % Check and print associated flag if it exists and is not empty
2547                        if ~isempty(flag) && eval(sprintf('exist(''%s'', ''var'') && ~isempty(%s)', flag, flag))
2548                            fprintf(fileId, '\t%2s\n', eval(flag));
2549                        end
2550                    end
2551                end
2552
2553                %Deactivate the Makefile section
2554                app.Mpif90EditField.Enable = 'off';
2555                app.Mpif90EditFieldLabel.Enable = 'off';
2556                app.Button_41.Enable = 'off';
2557                app.CreateMakefileButton.Enable = 'off';
2558                app.DoublePrecisionCheckBox.Enable = 'off';
2559                app.IntelCompilerCheckBox.Enable = 'off';
2560                app.ParallelModeCheckBox.Enable = 'off';
2561                app.MakefileEditField.Enable = 'off';
2562                app.MakefileEditFieldLabel.Enable = 'off';
2563                app.Button_42.Enable = 'off';
2564            else
2565                %Updating the log report
2566                fprintf(fileId, '%s\n', '');
2567                fprintf(fileId, '%s\n\n','> Status: The Makefile section is now enabled');
2568
2569                app.CreateMakefileButton.Enable = 'on';
2570                app.DoublePrecisionCheckBox.Enable = 'on';
2571                app.IntelCompilerCheckBox.Enable = 'on';
2572                app.ParallelModeCheckBox.Enable = 'on';
2573                app.MakefileEditField.Enable = 'on';
2574                app.MakefileEditFieldLabel.Enable = 'on';
2575                app.Button_42.Enable = 'on';
2576            end
2577
```

```
2578            %Close the log report and show it in the preview panel
2579            fclose(fileId);
2580            app.Preview_LogReport.Value = fileread(LogFile);
2581
2582            %Close the input.txt and show it in the preview panel
2583            fclose(fileInputId);
2584            fileContent = fileread(fileInputPath);
2585            app.Preview_Input.Value = fileContent;
2586
2587            drawnow
2588        end
2589
2590        % Callback function
2591        function ImportButton_4Pushed(app, event)
2592
2593            %
2594        end
2595
2596        % Value changed function: SimulationIntervalEditField
2597        function SimulationIntervalEditFieldValueChanged(app, event)
2598            app.SimulationIntervalEditField.FontColor = 'k';
2599        end
2600
2601        % Value changed function: GaugeIntervalEditField
2602        function GaugeIntervalEditFieldValueChanged(app, event)
2603            app.GaugeIntervalEditField.FontColor= 'k';
2604        end
2605
2606        % Value changed function: StartEditField
2607        function StartEditFieldValueChanged(app, event)
2608            app.StartEditField.FontColor = 'k';
2609
2610        end
2611
2612        % Value changed function: XEditField
2613        function XEditFieldValueChanged(app, event)
2614            app.XEditField.FontColor = 'k';
2615
2616        end
2617
2618        % Value changed function: YEditField
2619        function YEditFieldValueChanged(app, event)
2620            app.YEditField.FontColor = 'k';
2621        end
2622
2623        % Value changed function: LongitudeEditField
2624        function LongitudeEditFieldValueChanged(app, event)
2625            app.LongitudeEditField.FontColor = 'k';
2626
2627
2628        end
2629
2630        % Value changed function: LatitudeEditField
2631        function LatitudeEditFieldValueChanged(app, event)
2632            app.LatitudeEditField.FontColor = 'k';
2633        end
2634
2635        % Callback function
```

```
2636        function ButtonGroupSelectionChanged(app, event)
2637            selectedButton = app.ButtonGroup.SelectedObject;
2638            if selectedButton == app.SphericaldegreesButton
2639                app.SphericaldegreesButton.FontColor = 'k';
2640                app.CartesianmetersButton.FontColor = '0.50,0.50,0.50';
2641                value1 = app.XEditField;
2642                value2 = app.YEditField;
2643                value3 = '1.0';
2644                value4 = '1.0';
2645            elseif selectedButton == app.CartesianmetersButton
2646                app.CartesianmetersButton.FontColor = 'k';
2647                app.SphericaldegreesButton.FontColor = '0.50,0.50,0.50';
2648                value3 =  app.XEditField;
2649                value4=  app.YEditField;
2650                value1 = '1.0';
2651                value2 = '1.0';
2652            end
2653
2654            app.Dphi = value1;
2655            app.Dtheta = value2;
2656            app.DX = value3;
2657            app.DY = value4;
2658
2659        end
2660
2661        % Callback function
2662        function ButtonGroup_2SelectionChanged(app, event)
2663
2664            selectedButton = app.ButtonGroup_2.SelectedObject;
2665            app.SWE = selectedButton;
2666
2667        end
2668
2669        % Selection changed function: ButtonGroup_3
2670        function ButtonGroup_3SelectionChanged(app, event)
2671            app.FullyNonlinearBoussinesqEquationButton.FontColor = 'k';
2672            app.LinearShallowWaterEquationButton.FontColor = 'k';
2673            app.NonlinearShallowWaterEquationButton.FontColor = 'k';
2674
2675
2676            if app.FullyNonlinearBoussinesqEquationButton.Value
2677                app.ShockwavecapturingButton.Enable = "on";
2678                app.ViscositybreakingButton.Enable = "on";
2679                drawnow
2680                if   app.ViscositybreakingButton.Value
2681                    app.C2EditField.Enable = "on";
2682                    app.C2EditFieldLabel.Enable = "on";
2683                    app.C1EditField.Enable = "on";
2684                    app.C1EditFieldLabel.Enable = "on";
2685                end
2686            else
2687                app.ShockwavecapturingButton.Enable = "off";
2688                app.ViscositybreakingButton.Enable = "off";
2689                app.C2EditField.Enable = "off";
2690                app.C2EditFieldLabel.Enable = "off";
2691                app.C1EditField.Enable = "off";
2692                app.C1EditFieldLabel.Enable = "off";
2693            end
```

```
2694        end
2695
2696        % Selection changed function: ButtonGroup_4
2697        function ButtonGroup_4SelectionChanged(app, event)
2698            app.SphericalButton.FontColor = 'k';
2699            app.CartesianButton.FontColor = 'k';
2700
2701            if app.SphericalButton.Value
2702                app.LatitudeEditField.Enable = 'on';
2703                app.LatitudeEditFieldLabel.Enable = 'on';
2704                app.LongitudeEditField.Enable = 'on';
2705                app.LongitudeEditFieldLabel.Enable = 'on';
2706            else
2707                app.LatitudeEditField.Enable = 'off';
2708                app.LatitudeEditFieldLabel.Enable = 'off';
2709                app.LongitudeEditField.Enable = 'off';
2710                app.LongitudeEditFieldLabel.Enable = 'off';
2711                app.LatitudeEditField.FontColor = '0.65,0.65,0.65';
2712                app.LongitudeEditField.FontColor  = '0.65,0.65,0.65';
2713            end
2714
2715        end
2716
2717        % Value changed function: DepthCheckBox
2718        function DepthCheckBoxValueChanged(app, event)
2719
2720        end
2721
2722        % Value changed function: UVelocityCheckBox
2723        function UVelocityCheckBoxValueChanged(app, event)
2724
2725
2726        end
2727
2728        % Value changed function: VVelocityCheckBox
2729        function VVelocityCheckBoxValueChanged(app, event)
2730
2731        end
2732
2733        % Value changed function: SurfaceElevationattimetCheckBox
2734        function SurfaceElevationattimetCheckBoxValueChanged(app, event)
2735
2736        end
2737
2738        % Value changed function: WetdrymaskCheckBox
2739        function WetdrymaskCheckBoxValueChanged(app, event)
2740
2741        end
2742
2743        % Value changed function: WetdrymaskforBoussinesqNSWECheckBox
2744        function WetdrymaskforBoussinesqNSWECheckBoxValueChanged(app, event)
2745
2746        end
2747
2748        % Value changed function: XSourceCheckBox
2749        function XSourceCheckBoxValueChanged(app, event)
2750
2751        end
```

```
2752
2753        % Value changed function: YSourceCheckBox
2754        function YSourceCheckBoxValueChanged(app, event)
2755
2756        end
2757
2758        % Value changed function: PMomentumFluxCheckBox
2759        function PMomentumFluxCheckBoxValueChanged(app, event)
2760
2761        end
2762
2763        % Value changed function: QmomentumFluxCheckBox
2764        function QmomentumFluxCheckBoxValueChanged(app, event)
2765
2766        end
2767
2768        % Value changed function: FxFluxCheckBox
2769        function FxFluxCheckBoxValueChanged(app, event)
2770
2771        end
2772
2773        % Value changed function: FyFluxCheckBox
2774        function FyFluxCheckBoxValueChanged(app, event)
2775
2776        end
2777
2778        % Value changed function: GxFluxCheckBox
2779        function GxFluxCheckBoxValueChanged(app, event)
2780
2781        end
2782
2783        % Value changed function: GyFluxCheckBox
2784        function GyFluxCheckBoxValueChanged(app, event)
2785
2786        end
2787
2788        % Value changed function: BreakingAgeCheckBox
2789        function BreakingAgeCheckBoxValueChanged(app, event)
2790
2791        end
2792
2793        % Value changed function: MaximumWaveHeightCheckBox
2794        function MaximumWaveHeightCheckBoxValueChanged(app, event)
2795
2796        end
2797
2798        % Value changed function: MinimumWaveHeightCheckBox
2799        function MinimumWaveHeightCheckBoxValueChanged(app, event)
2800
2801        end
2802
2803        % Value changed function: MaximumVelocityCheckBox
2804        function MaximumVelocityCheckBoxValueChanged(app, event)
2805
2806        end
2807
2808        % Value changed function: UndertowCheckBox
2809        function UndertowCheckBoxValueChanged(app, event)
```

```
2810
2811        end
2812
2813        % Value changed function: MaxVorticityCheckBox
2814        function MaxVorticityCheckBoxValueChanged(app, event)
2815
2816        end
2817
2818        % Value changed function: MaxMomentumFluxCheckBox
2819        function MaxMomentumFluxCheckBoxValueChanged(app, event)
2820
2821        end
2822
2823        % Value changed function: ArrivalCheckBox
2824        function ArrivalCheckBoxValueChanged(app, event)
2825        if app.ArrivalCheckBox.Value
2826            app.ThresholdEditFieldLabel.Visible = "on";
2827            app.ThresholdEditField.Visible = "on";
2828        else
2829            app.ThresholdEditFieldLabel.Visible = "off";
2830            app.ThresholdEditField.Visible = "off";
2831            app.ThresholdEditField.FontColor = '0.65,0.65,0.65';
2832        end
2833
2834
2835        end
2836
2837        % Value changed function: WaveHeightCheckBox
2838        function WaveHeightCheckBoxValueChanged(app, event)
2839
2840        end
2841
2842        % Value changed function: RollerinducedFluxCheckBox
2843        function RollerinducedFluxCheckBoxValueChanged(app, event)
2844
2845        end
2846
2847        % Value changed function: PressureFieldCheckBox
2848        function PressureFieldCheckBoxValueChanged(app, event)
2849
2850        end
2851
2852        % Value changed function: BreakingLocationCheckBox
2853        function BreakingLocationCheckBoxValueChanged(app, event)
2854
2855        end
2856
2857        % Callback function
2858        function EastlimitmEditFieldValueChanged(app, event)
2859            app.EastEditField.FontColor = 'k';
2860        end
2861
2862        % Callback function
2863        function WestbountmEditFieldValueChanged(app, event)
2864            app.WestEditField.FontColor = 'k';
2865        end
2866
2867        % Value changed function: SouthEditField
```

```
2868        function SouthEditFieldValueChanged(app, event)
2869            app.SouthEditField.FontColor = 'k';
2870        end
2871
2872        % Value changed function: NorthEditField
2873        function NorthEditFieldValueChanged(app, event)
2874            app.NorthEditField.FontColor = 'k';
2875        end
2876
2877        % Callback function
2878        function ButtonGroup_5SelectionChanged(app, event)
2879
2880        end
2881
2882        % Callback function
2883        function DirectSpongeSwitchValueChanged(app, event)
2884
2885             if strcmp(app.DirectSpongeSwitch.Value,'On')
2886                app.DecayTypeDropDown.Visible = "on";
2887                app.DecayTypeDropDownLabel.Visible = "on";
2888                app.Panel_16.Enable = "on";
2889             elseif strcmp(app.DirectSpongeSwitch.Value,'Off')
2890                app.DecayTypeDropDown.Visible = "off";
2891                app.DecayTypeDropDownLabel.Visible = "off";
2892             end
2893
2894
2895
2896        end
2897
2898        % Callback function
2899        function Button_9Pushed(app, event)
2900               maindir = uigetdir;
2901               maindir = fullfile(maindir,' ');
2902               app.ResultsfolderEditField.Value = deblank(string(maindir));
2903               app.ResultsfolderEditField.FontColor = 'k';
2904        end
2905
2906        % Button pushed function: Button_7
2907        function Button_7Pushed2(app, event)
2908            [filename3,path3] = uigetfile('*.txt;*.shp');
2909            fullname = fullfile(path3,filename3);
2910            try
2911            [~, name, ~] = fileparts(filename3);
2912            app.GaugeFname = name;
2913            app.LoadFileEditField.Value = filename3;
2914            app.GAUGEFILE = filename3;
2915            app.GAUGEFILElocation = fullname;
2916            app.LoadFileEditField.FontColor = 'k';
2917            app.LoadFileEditField.BackgroundColor = 'w';
2918
2919            if contains(filename3,'.shp')
2920                gauge = shaperead(fullname);
2921                latGT = [];
2922                lonGT = [];
2923                app.latGauge = latGT;
2924
2925                for k = 1:length(gauge)
```

```
2926                    latGT = [latGT, gauge(k).X];   % Concatenate X coordinates
2927                    lonGT = [lonGT, gauge(k).Y];   % Concatenate Y coordinates
2928                end
2929
2930                list = [round(lonGT,4)',round(latGT,4)'];
2931                list =  unique(list,'rows'); %remove duplicate lat-long↙
combination
2932                app.NumberStations = num2str(height(list));
2933                app.STATION_FILE = string(name) + ".txt";
2934
2935                %Save as text file
2936                app.GaugeListGrids = list;
2937
2938            elseif contains(filename3,'.txt')
2939                gaugetext = readmatrix(fullname);
2940                app.gaugetext0 = gaugetext;
2941                app.NumberStations = num2str(height(gaugetext));
2942                app.STATION_FILE = string(filename3);
2943
2944            end
2945
2946            catch
2947                %do nothing
2948            end
2949
2950        end
2951
2952        % Value changed function: EastEditField
2953        function EastEditFieldValueChanged(app, event)
2954            app.EastEditField.FontColor = 'k';
2955        end
2956
2957        % Value changed function: WestEditField
2958        function WestEditFieldValueChanged(app, event)
2959            app.WestEditField.FontColor = 'k';
2960        end
2961
2962        % Value changed function: ResultsfolderEditField
2963        function ResultsfolderEditFieldValueChanged(app, event)
2964
2965
2966        end
2967
2968        % Value changed function: ProcessorX
2969        function ProcessorXValueChanged(app, event)
2970            app.ProcessorX.FontColor = 'k';
2971
2972        end
2973
2974        % Value changed function: ProcessorY
2975        function ProcessorYValueChanged(app, event)
2976            app.ProcessorY.FontColor = 'k';
2977
2978        end
2979
2980        % Value changed function: TotalEditField
2981        function TotalEditFieldValueChanged2(app, event)
2982            app.TotalEditField.FontColor = 'k';
```

```
2983
2984        end
2985
2986        % Button pushed function: Button_32
2987        function Button_4Pushed(app, event)
2988                [bathymetryfile,path2] = uigetfile('*.*');
2989                app.SurfaceHeightEditField.Value = string(bathymetryfile);
2990                app.SurfaceHeightEditField.FontColor = 'k';
2991                app.SurfaceHeightEditField.BackgroundColor = 'w';
2992                app.SurfaceHeightLOC = string(path2) + string(bathymetryfile);
2993                app.SurfaceHeightNAME = string(bathymetryfile);
2994
2995
2996        end
2997
2998        % Button pushed function: Button_34
2999        function Button_5Pushed(app, event)
3000            [file,path2] = uigetfile('*.*');
3001                app.UVelocityEditField.Value = string(file);
3002                app.UVelocityEditField.FontColor = 'k';
3003                app.UVelocityEditField.BackgroundColor = 'w';
3004                app.XVelocityLOC = string(path2) + string(file);
3005                app.XVelocityNAME = string(file);
3006
3007        end
3008
3009        % Button pushed function: Button_33
3010        function Button_6Pushed(app, event)
3011                [file,path2] = uigetfile('*.*');
3012                app.VVelocityEditField.Value = string(file);
3013                app.VVelocityEditField.FontColor = 'k';
3014                app.VVelocityEditField.BackgroundColor = 'w';
3015                app.YVelocityLOC = string(path2) + string(file);
3016                app.YVelocityNAME = string(file);
3017        end
3018
3019        % Selection changed function: ButtonGroup_18
3020        function ButtonGroup_6SelectionChanged(app, event)
3021
3022                    if  app.ImportUVZButton.Value
3023                        app.TabGroup2.SelectedTab =   app.Tab;
3024
3025                    end
3026
3027                     if app.WavemakerButton.Value
3028                        app.TabGroup2.SelectedTab = app.Tab_2;
3029
3030                     end
3031
3032                    if app.MeteotsunamiButton.Value
3033                         app.TabGroup2.SelectedTab = app.Tab_4;
3034                         app.MinimumWaveHeightCheckBox.Value = 1;
3035                         app.PressureFieldCheckBox.Value = 1;
3036                    else
3037                        %uncheck the expected output values
3038                         app.MinimumWaveHeightCheckBox.Value = 0;
3039                         app.PressureFieldCheckBox.Value = 0;
3040                    end
```

```
3041
3042
3043                    if app.VesselButton.Value
3044                        app.TabGroup2.SelectedTab = app.Tab_6;
3045
3046                    end
3047
3048        end
3049
3050        % Value changed function: DepthFileEditField
3051        function DepthFileEditFieldValueChanged(app, event)
3052          app.DepthFileEditField.BackgroundColor = 'w';
3053          app.DepthFileEditField.FontColor = 'k';
3054
3055        end
3056
3057        % Selection changed function: ButtonGroup_7
3058        function ButtonGroup_7SelectionChanged(app, event)
3059            %List all the items to be displayed or hidden
3060            function setVisibility(app, depthVisible, angleVisible,↙
startXVisible, button8Visible, depthFileVisible, waterLevelVisible,↙
invertValuesVisible)
3061                app.DepthmEditFieldLabel.Visible = depthVisible;
3062                app.DepthmEditField.Visible = depthVisible;
3063                app.AngleEditFieldLabel.Visible = angleVisible;
3064                app.AngleEditField.Visible = angleVisible;
3065                app.StartatXEditFieldLabel.Visible = startXVisible;
3066                app.StartatXEditField.Visible = startXVisible;
3067                app.Button_8.Visible = button8Visible;
3068                app.DepthFileEditField.Visible = depthFileVisible;
3069                app.WaterlevelmEditFieldLabel.Visible = waterLevelVisible;
3070                app.WaterlevelmEditField.Visible = waterLevelVisible;
3071                app.InvertvaluesCheckBox.Visible = invertValuesVisible;
3072            end
3073
3074
3075            if app.ImportDataButton.Value
3076                setVisibility(app, "off", "off", "off", "on", "on", "on",↙
"on");
3077            elseif app.FlatButton.Value
3078                setVisibility(app, "on", "off", "off", "off", "off", "off",↙
"off");
3079            elseif app.SlopeButton.Value
3080                setVisibility(app, "on", "on", "on", "off", "off", "off",↙
"off");
3081            end
3082
3083            % Set font color to black
3084            if ~app.Button4.Value
3085                app.ImportDataButton.FontColor = 'k';
3086                app.SlopeButton.FontColor = 'k';
3087                app.FlatButton.FontColor = 'k';
3088            end
3089
3090        end
3091
3092        % Size changed function: InputTab
3093        function InputTabSizeChanged(app, event)
```

```
3094
3095        end
3096
3097        % Value changed function: DepthmEditField
3098        function DepthmEditFieldValueChanged(app, event)
3099
3100            app.DepthmEditField.FontColor = 'k';
3101            app.DepthmEditField.BackgroundColor = 'w';
3102
3103        end
3104
3105        % Value changed function: AngleEditField
3106        function AngleEditFieldValueChanged(app, event)
3107            app.AngleEditField.FontColor = 'k';
3108            app.AngleEditField.BackgroundColor = 'w';
3109        end
3110
3111        % Value changed function: StartatXEditField
3112        function StartatXEditFieldValueChanged(app, event)
3113           app.StartatXEditField.FontColor = 'k';
3114           app.StartatXEditField.BackgroundColor = 'w';
3115
3116        end
3117
3118        % Value changing function: ResultsfolderEditField
3119        function ResultsfolderEditFieldValueChanging(app, event)
3120
3121            app.ResultsfolderEditField.BackgroundColor = 'w';
3122            app.ResultsfolderEditField.FontColor = 'k';
3123        end
3124
3125        % Value changed function: YEditField_2
3126        function YEditField_2ValueChanged(app, event)
3127           app.YEditField_2.FontColor = 'k';
3128
3129        end
3130
3131        % Button pushed function: Button_10
3132        function Button_10Pushed(app, event)
3133               [frictionfile,frictionpath] = uigetfile('*.txt;*.tif');
3134               app.FrictionFileEditField.Value = string(frictionfile);
3135               app.Cd_file = fullfile(string(frictionpath) + string↙
(frictionfile));
3136               app.filenamefriction = string(frictionfile);
3137               app.FrictionFileEditField.FontColor = 'k';
3138               app.FrictionFileEditField.BackgroundColor = 'w';
3139        end
3140
3141        % Value changing function: FrictionFileEditField
3142        function FrictionFileEditFieldValueChanging(app, event)
3143                app.FrictionFileEditField.Value = '';
3144                app.FrictionFileEditField.FontColor = 'k';
3145                app.FrictionFileEditField.BackgroundColor = 'w';
3146        end
3147
3148        % Selection changed function: ButtonGroup_8
3149        function ButtonGroup_8SelectionChanged(app, event)
3150            if app.ConstantCoefficientButton.Value
```

```
3151                 app.FrictionFileEditField.Visible = "off";
3152                 app.FrictionFileEditFieldLabel.Visible = "off";
3153                 app.Button_10.Visible = "off";
3154                 app.ConstantFrictionValue.Visible = "on";
3155                 app.ValueEditFieldLabel.Visible = "on";
3156 %                 app.BottomFrictionEditField.Enable = "on";
3157 %                 app.BottomFrictionEditFieldLabel.Enable  = "on";
3158             elseif app.NonconstantButton.Value
3159                 app.FrictionFileEditField.Visible = "on";
3160                 app.FrictionFileEditFieldLabel.Visible = "on";
3161                 app.Button_10.Visible = "on";
3162                 app.ConstantFrictionValue.Visible = "off";
3163                 app.ValueEditFieldLabel.Visible = "off";
3164 %                 app.BottomFrictionEditField.Enable = "on";
3165 %                 app.BottomFrictionEditFieldLabel.Enable  = "on";
3166             end
3167 
3168         end
3169 
3170         % Value changed function: ConstantFrictionValue
3171         function ConstantFrictionValueValueChanged(app, event)
3172             app.ConstantFrictionValue.FontColor = 'k';
3173             app.ConstantFrictionValue.BackgroundColor = 'w';
3174         end
3175 
3176         % Value changed function: C2EditField
3177         function C2EditFieldValueChanged(app, event)
3178              app.C2EditField.FontColor = 'k';
3179 
3180 
3181         end
3182 
3183         % Value changed function: C1EditField
3184         function C1EditFieldValueChanged(app, event)
3185 
3186              app.C1EditField.FontColor = 'k';
3187         end
3188 
3189         % Value changed function: WetDrySchemeEditField
3190         function WetDrySchemeEditFieldValueChanged(app, event)
3191             app.WetDrySchemeEditField.FontColor = 'k';
3192 
3193 
3194         end
3195 
3196         % Value changed function: BottomFrictionEditField
3197         function BottomFrictionEditFieldValueChanged(app, event)
3198             app.BottomFrictionEditField.FontColor = 'k';
3199 
3200         end
3201 
3202         % Value changing function: SavefilestoEditField
3203         function SavefilestoEditFieldValueChanging(app, event)
3204             app.SavefilestoEditField.Value = '';
3205             app.SavefilestoEditField.FontColor = 'k';
3206             app.SavefilestoEditField.BackgroundColor = 'w';
3207 
3208         end
```

```
3209
3210         % Value changed function: DecayTypeDropDown
3211         function DecayTypeDropDownValueChanged(app, event)
3212
3213         end
3214
3215         % Button pushed function: ObstacleLoadFile
3216         function Button_19Pushed(app, event)
3217              [filename,path] = uigetfile('*.txt;*.tif');
3218              app.ObstacleFilename.Value = string(filename);
3219              app.ObstacleFilename.FontColor = 'k';
3220              app.ObstacleFilename.BackgroundColor = 'w';
3221              app.obstacleLOC = string(path) + string(filename);
3222              app.obstacleNAME = string(filename);
3223         end
3224
3225         % Value changed function: MaximumCdEditField
3226         function MaximumCdEditFieldValueChanged(app, event)
3227             app.MaximumCdEditField.FontColor = 'k';
3228
3229         end
3230
3231         % Value changed function: MaximumCspEditField
3232         function MaximumCspEditFieldValueChanged(app, event)
3233          app.MaximumCspEditField.FontColor = 'k';
3234
3235         end
3236
3237         % Value changed function: WavemakertypeDropDown
3238         function WavemakertypeDropDownValueChanged(app, event)
3239 %Disable and reset all fields and buttons first
3240 allFields = [app.XcoordinatemEditField, app.YcoordinatemEditField, app.↙
WidthmEditField, app.WidthDeltaEditField, app.AmplitudemEditField, app.↙
WaterdepthmEditField, app.LagtimesecEditField, app.ThetadegreesEditField, app.↙
TimerampsecEditField, app.PeriodsecEditField, app.WavecomponentfileEditField, app.↙
WaveCompCountEditField,app.MinimumEditField,app.MaximumEditField,app.PeakEditField,↙
app.WaveHeightmEditField];
3241 for field = allFields
3242     field.Enable = 'off';
3243     field.FontColor = [0.8, 0.8, 0.8];
3244     if isprop(field, 'Value') && isnumeric(field.Value)
3245         field.Value = 0;
3246     elseif isprop(field, 'Value')
3247         field.Value = "";
3248     end
3249 end
3250 app.Button_20.Enable = 'off'; %Disable import buttom of 'Save Comoponent File'
3251
3252
3253 %List of the paremeters to be enabled based on the wavemaker type selected
3254 wavemakerSettings = struct(...
3255     'INI_REC', {{app.XcoordinatemEditField, app.YcoordinatemEditField, app.↙
WidthmEditField}}, ...
3256     'LEF_SOL', {{app.AmplitudemEditField, app.WaterdepthmEditField, app.↙
LagtimesecEditField}}, ...
3257     'INI_SOL', {{app.AmplitudemEditField, app.WaterdepthmEditField, app.↙
XcoordinatemEditField}}, ...
3258     'WK IRR', {{app.XcoordinatemEditField,app.YcoordinatemEditField,app.↙
```

```
WidthmEditField,app.TimerampsecEditField,app.WidthDeltaEditField,app.
MinimumEditField,app.MaximumEditField,app.PeakEditField,app.WaveHeightmEditField,
app.WaterdepthmEditField}}, ...
    'JON_1D', {{app.XcoordinatemEditField,app.YcoordinatemEditField,app.
WidthmEditField,app.TimerampsecEditField,app.WidthDeltaEditField,app.
MinimumEditField,app.MaximumEditField,app.PeakEditField,app.WaveHeightmEditField,
app.WaterdepthmEditField}}, ...
    'JON_2D', {{app.XcoordinatemEditField,app.YcoordinatemEditField,app.
WidthmEditField,app.TimerampsecEditField,app.WidthDeltaEditField,app.
MinimumEditField,app.MaximumEditField,app.PeakEditField,app.WaveHeightmEditField,
app.WaterdepthmEditField}}, ...
    'TMA_1D', {{app.XcoordinatemEditField,app.YcoordinatemEditField,app.
WidthmEditField,app.TimerampsecEditField,app.WidthDeltaEditField,app.
MinimumEditField,app.MaximumEditField,app.PeakEditField,app.WaveHeightmEditField,
app.WaterdepthmEditField}}, ...
    'WK_REG', {{app.XcoordinatemEditField, app.AmplitudemEditField, app.
WaterdepthmEditField, app.ThetadegreesEditField, app.TimerampsecEditField,app.
WidthDeltaEditField}}, ...
    'WK_TIME_SERIES', {{app.WavecomponentfileEditField, app.Button_20, app.
PeriodsecEditField, app.WaterdepthmEditField, app.XcoordinatemEditField, app.
WidthmEditField, app.WaveCompCountEditField}}, ...
    'WK_DATA2D', {{app.XcoordinatemEditField, app.YcoordinatemEditField, app.
WidthDeltaEditField, app.WaterdepthmEditField, app.WavecomponentfileEditField, app.
Button_20}}, ...
    'INI_GAUSSIAN', {{app.AmplitudemEditField, app.XcoordinatemEditField, app.
YcoordinatemEditField, app.WidthmEditField}} ...
);

%Enable paremeters based on the wavemaker type selected
selectedType = app.WavemakertypeDropDown.Value;
if isfield(wavemakerSettings, selectedType)
    fieldsToEnable = wavemakerSettings.(selectedType);
    for fieldCell = fieldsToEnable
        field = fieldCell{:};
        field.Enable = 'on';
    end
end

if ~strcmp(app.WavemakertypeDropDown.Value, 'WK_TIME_SERIES')
    app.WavecomponentfileEditField.BackgroundColor = 'w';
    app.WavecomponentfileEditField.Value = '';
    app.WavecomponentfileEditField.FontColor = 'k';
end

        end

        % Drop down opening function: WavemakertypeDropDown
        function WavemakertypeDropDownOpening(app, event)
            app.WavemakertypeDropDown.FontColor = '0.00,0.00,0.00';
        end

        % Value changed function: AmplitudemEditField
        function AmplitudemEditFieldValueChanged(app, event)
            app.AmplitudemEditField.FontColor = 'k';
        end

        % Value changed function: WaterdepthmEditField
        function WaterdepthmEditFieldValueChanged(app, event)
```

```
3298            app.WaterdepthmEditField.FontColor = 'k';
3299
3300
3301        end
3302
3303        % Value changed function: LagtimesecEditField
3304        function LagtimesecEditFieldValueChanged(app, event)
3305            app.LagtimesecEditField.FontColor = 'k';
3306
3307        end
3308
3309        % Value changed function: XcoordinatemEditField
3310        function XcoordinatemEditFieldValueChanged(app, event)
3311           app.XcoordinatemEditField.FontColor = 'k';
3312
3313
3314        end
3315
3316        % Callback function
3317        function XcentercoordinateEditFieldValueChanged(app, event)
3318
3319
3320        end
3321
3322        % Value changed function: YcoordinatemEditField
3323        function YcoordinatemEditFieldValueChanged(app, event)
3324            app.YcoordinatemEditField.FontColor = 'k';
3325
3326
3327        end
3328
3329        % Value changed function: WidthmEditField
3330        function WidthmEditFieldValueChanged(app, event)
3331            app.WidthmEditField.FontColor = 'k';
3332
3333
3334        end
3335
3336        % Value changed function: TimerampsecEditField
3337        function TimerampsecEditFieldValueChanged(app, event)
3338            app.TimerampsecEditField.FontColor = 'k';
3339
3340
3341        end
3342
3343        % Value changed function: WidthDeltaEditField
3344        function WidthDeltaEditFieldValueChanged(app, event)
3345            app.WidthDeltaEditField.FontColor = 'k';
3346
3347
3348        end
3349
3350        % Value changed function: PeriodsecEditField
3351        function PeriodsecEditFieldValueChanged(app, event)
3352            app.PeriodsecEditField.FontColor = 'k';
3353
3354
3355        end
```

```
3356
3357        % Button pushed function: Button_20
3358        function Button_20Pushed(app, event)
3359              [file,path] = uigetfile('*.*'); %*.txt;
3360                app.WavecomponentfileEditField.Value = string(file);
3361                app.WaveFileNAME = string(file);
3362                app.WaveFileLOC = string(path) + string(file);
3363                app.WavecomponentfileEditField.BackgroundColor = 'w';
3364                app.WavecomponentfileEditField.FontColor =  'k';
3365
3366        end
3367
3368        % Value changed function: WavecomponentfileEditField
3369        function WavecomponentfileEditFieldValueChanged(app, event)
3370
3371
3372        end
3373
3374        % Value changed function: ThetadegreesEditField
3375        function ThetadegreesEditFieldValueChanged(app, event)
3376            app.ThetadegreesEditField.FontColor = 'k';
3377
3378        end
3379
3380        % Value changed function: XEditField_2
3381        function XEditField_2ValueChanged(app, event)
3382           app.XEditField_2.FontColor = 'k';
3383
3384
3385        end
3386
3387        % Callback function
3388        function TimeintervalsecEditFieldValueChanged(app, event)
3389            app.TimeintervalsecEditField.FontColor = 'k';
3390
3391        end
3392
3393        % Callback function
3394        function SteadyTimeEditFieldValueChanged(app, event)
3395            app.SteadyTimeEditField.FontColor = 'k';
3396
3397        end
3398
3399        % Callback function
3400        function ButtonGroup_12SelectionChanged(app, event)
3401
3402
3403        end
3404
3405        % Callback function
3406        function SchemeDropDownValueChanged(app, event)
3407
3408        end
3409
3410        % Value changing function: UVelocityEditField
3411        function UVelocityEditFieldValueChanging(app, event)
3412
3413        end
```

```
3414
3415        % Value changing function: SurfaceHeightEditField
3416        function SurfaceHeightEditFieldValueChanging(app, event)
3417
3418
3419        end
3420
3421        % Callback function
3422        function Button5Pushed(app, event)
3423            app.TabGroup2.SelectedTab =   app.Tab_3;        %app.InputsTab;
3424        end
3425
3426        % Value changing function: VVelocityEditField
3427        function VVelocityEditFieldValueChanging(app, event)
3428
3429
3430        end
3431
3432        % Value changed function: WindforceSwitch
3433        function WindforceSwitchValueChanged(app, event)
3434              if strcmp(app.WindforceSwitch.Value,"On")
3435                  app.WindwaveinteractionSwitch.Enable = "on";
3436                  app.WindwaveinteractionSwitchLabel.Enable = "on";
3437                  app.WindstresscoefficientEditField.Enable = "on";
3438                  app.WindstresscoefficientEditFieldLabel.Enable = "on";
3439                  app.CrestRatioEditField.Enable = "on";
3440                  app.CrestRatioEditFieldLabel.Enable ="on";
3441
3442              elseif strcmp(app.WindforceSwitch.Value,"Off")
3443                  app.WindwaveinteractionSwitch.Enable = "off";
3444                  app.WindwaveinteractionSwitchLabel.Enable = "off";
3445                  app.WindstresscoefficientEditField.Enable = "off";
3446                  app.WindstresscoefficientEditFieldLabel.Enable = "off";
3447                  app.CrestRatioEditField.Enable = "off";
3448                  app.CrestRatioEditFieldLabel.Enable ="off";
3449                  app.WindstresscoefficientEditField.FontColor =↙
'0.65,0.65,0.65';
3450                  app.WindstresscoefficientEditField.FontColor =↙
'0.65,0.65,0.65';
3451                  app.CrestRatioEditField.FontColor = '0.65,0.65,0.65';
3452              end
3453
3454        end
3455
3456        % Value changed function: ConstantwindfieldSwitch
3457        function ConstantwindfieldSwitchValueChanged(app, event)
3458
3459            if strcmp(app.ConstantwindfieldSwitch.Value,"On")
3460                app.ConstantwindfileEditFieldLabel.Enable = "on";
3461                app.ConstantwindfileEditField.Enable = "on";
3462                app.Button_29.Enable = "on";
3463                app.WindforceSwitch.Value = "On";
3464
3465                app.ConstantwindfieldSwitch.Enable = "on";
3466                app.ConstantwindfieldSwitchLabel.Enable = "on";
3467                app.WindwaveinteractionSwitch.Enable = "on";
3468                app.WindwaveinteractionSwitchLabel.Enable = "on";
3469                app.WindstresscoefficientEditField.Enable = "on";
```

```
3470                 app.WindstresscoefficientEditFieldLabel.Enable = "on";
3471                 app.CrestRatioEditField.Enable = "on";
3472                 app.CrestRatioEditFieldLabel.Enable ="on";
3473 
3474             elseif strcmp(app.ConstantwindfieldSwitch.Value,"Off")
3475                 app.ConstantwindfileEditFieldLabel.Enable = "off";
3476                 app.ConstantwindfileEditField.Enable = "off";
3477                 app.Button_29.Enable = "off";
3478                 app.WindforceSwitch.Value = "Off";
3479                 app.WindwaveinteractionSwitch.Enable = "off";
3480                 app.WindwaveinteractionSwitchLabel.Enable = "off";
3481                 app.WindstresscoefficientEditField.Enable = "off";
3482                 app.WindstresscoefficientEditFieldLabel.Enable = "off";
3483                 app.CrestRatioEditField.Enable = "off";
3484                 app.CrestRatioEditFieldLabel.Enable ="off";
3485             end
3486 
3487         end
3488 
3489         % Button pushed function: Button_29
3490         function Button_29Pushed(app, event)
3491             [file,path2] = uigetfile('*.*');
3492             app.ConstantwindfileEditField.Value = string(file);
3493             app.ConstantwindfileEditField.FontColor = 'k';
3494             app.ConstantwindfileEditField.BackgroundColor = 'w';
3495                 app.ConsWindLOC = string(path2) + string(file);
3496                 app.ConsWindNAME = string(file);
3497         end
3498 
3499         % Value changed function: CrestRatioEditField
3500         function CrestRatioEditFieldValueChanged(app, event)
3501 
3502         end
3503 
3504         % Value changed function: WindstresscoefficientEditField
3505         function WindstresscoefficientEditFieldValueChanged(app, event)
3506            app.WindstresscoefficientEditField.FontColor =  'k';
3507 
3508         end
3509 
3510         % Button pushed function: Button_30
3511         function Button_30Pushed(app, event)
3512               [file,path2] = uigetfile('*.*');
3513                 app.StormfileEditField.Value = string(file);
3514                 app.StormfileEditField.FontColor = 'k';
3515                 app.StormfileEditField.BackgroundColor = 'w';
3516                 app.StormFileLOC = string(path2) + string(file);
3517                 app.StormFileNAME = string(file);
3518         end
3519 
3520         % Callback function
3521         function Button_39Pushed(app, event)
3522                [file,path2] = uigetfile('*.*');
3523                 app.PressurefileEditField.Value = string(file);
3524                 app.PressurefileEditField.FontColor = 'k';
3525                 app.PressurefileEditField.BackgroundColor = 'w';
3526                 app.PressureFileLOC = string(path2) + string(file);
3527                 app.PressureFileNAME = string(file);
```

```
3528        end
3529
3530        % Value changed function: WindwaveinteractionSwitch
3531        function WindwaveinteractionSwitchValueChanged(app, event)
3532
3533
3534        end
3535
3536        % Value changed function: AirpressureSwitch
3537        function AirpressureSwitchValueChanged(app, event)
3538            if strcmp(app.AirpressureSwitch.Value,"On")
3539                app.PressurefileEditField.Enable = "on";
3540                app.PressurefileEditFieldLabel.Enable = "on";
3541                app.Button_43.Enable = "on";
3542            else
3543                app.PressurefileEditField.Enable = "off";
3544                app.PressurefileEditFieldLabel.Enable = "off";
3545                app.Button_43.Enable = "off";
3546            end
3547        end
3548
3549        % Value changed function: HollandstormmodelSwitch
3550        function HollandstormmodelSwitchValueChanged(app, event)
3551            if strcmp(app.HollandstormmodelSwitch.Value,"On")
3552                app.StormfileEditFieldLabel.Enable = "on";
3553                app.StormfileEditField.Enable = "on";
3554                app.Button_30.Enable = "on";
3555            else
3556                app.StormfileEditFieldLabel.Enable = "off";
3557                app.StormfileEditField.Enable = "off";
3558                app.Button_30.Enable = "off";
3559            end
3560        end
3561
3562        % Value changed function: MediangrainDMEditField
3563        function MediangrainDMEditFieldValueChanged(app, event)
3564            app.MediangrainDMEditField.FontColor = 'k';
3565
3566        end
3567
3568        % Value changed function: SedimentdensityEditField
3569        function SedimentdensityEditFieldValueChanged(app, event)
3570            app.SedimentdensityEditField.FontColor = 'k';
3571
3572        end
3573
3574        % Value changed function: SedimentporosityEditField
3575        function SedimentporosityEditFieldValueChanged(app, event)
3576            app.SedimentporosityEditField.FontColor = 'k';
3577
3578        end
3579
3580        % Value changed function: SettlingvelocitymsEditField
3581        function SettlingvelocitymsEditFieldValueChanged(app, event)
3582            app.SettlingvelocitymsEditField.FontColor = 'k';
3583
3584        end
3585
```

```
3586        % Value changed function: CriticalShieldsEditField
3587        function CriticalShieldsEditFieldValueChanged(app, event)
3588            app.CriticalShieldsEditField.FontColor = 'k';
3589
3590        end
3591
3592        % Value changed function: ShieldsparamBedloadEditField
3593        function ShieldsparamBedloadEditFieldValueChanged(app, event)
3594            app.ShieldsparamBedloadEditField.FontColor = 'k';
3595
3596        end
3597
3598        % Value changed function: RungeKuttaparameter1EditField
3599        function RungeKuttaparameter1EditFieldValueChanged(app, event)
3600            app.RungeKuttaparameter1EditField.FontColor = 'k';
3601
3602        end
3603
3604        % Value changed function: RungeKuttaparameter2EditField
3605        function RungeKuttaparameter2EditFieldValueChanged(app, event)
3606            app.RungeKuttaparameter2EditField.FontColor = 'k';
3607
3608        end
3609
3610        % Value changed function: MinDepthSedPickUpEditField
3611        function MinDepthSedPickUpEditFieldValueChanged(app, event)
3612            app.MinDepthSedPickUpEditField.FontColor = 'k';
3613
3614        end
3615
3616        % Value changed function: TimeStepSedToMorphlEditField
3617        function TimeStepSedToMorphlEditFieldValueChanged(app, event)
3618            app.TimeStepSedToMorphlEditField.FontColor = 'k';
3619
3620        end
3621
3622        % Callback function
3623        function PressurefileEditFieldValueChanged(app, event)
3624            app.PressurefileEditField.FontColor = 'k';
3625
3626        end
3627
3628        % Value changed function: TimeStepBouss
3629        function TimeStepBoussValueChanged(app, event)
3630           app.TimeStepBouss.FontColor = 'k';
3631
3632        end
3633
3634        % Value changed function: RungeKuttaparam2EditField_2
3635        function RungeKuttaparam2EditField_2ValueChanged(app, event)
3636           app.RungeKuttaparam2EditField_2.FontColor = 'k';
3637
3638        end
3639
3640        % Button pushed function: Button_35
3641        function Button_35Pushed(app, event)
3642            [file,path2] = uigetfile('*.*');
3643            app.FileEditField.Value = string(file);
```

```
3644                 app.FileEditField.FontColor = 'k';
3645                 app.FileEditField.BackgroundColor = 'w';
3646                 app.HardBottomFileLOC = string(path2) + string(file);
3647                 app.HardBottomFileNAME = string(file);
3648         end
3649
3650         % Selection changed function: ButtonGroup_21
3651         function ButtonGroup_21SelectionChanged(app, event)
3652            if app.HardbednoerosionButton.Value
3653                app.Button_35.Visible= "on";
3654                app.FileEditField.Visible = "on";
3655                app.FileEditFieldLabel.Visible = "on";
3656            elseif app.SedimentbedButton.Value
3657
3658                app.Button_35.Visible = "off";
3659                app.FileEditField.Visible = "off";
3660                app.FileEditFieldLabel.Visible = "off";
3661            end
3662
3663            if ~app.Button_37.Value
3664                app.SedimentbedButton.FontColor = 'k';
3665                app.HardbednoerosionButton.FontColor = 'k';
3666            end
3667
3668         end
3669
3670         % Selection changed function: ButtonGroup_20
3671         function ButtonGroup_20SelectionChanged(app, event)
3672             if ~app.Button_36.Value
3673                 app.NobedchangeButton.FontColor = 'k';
3674                 app.UpdatedepthButton.FontColor = 'k';
3675             end
3676         end
3677
3678         % Selection changed function: ButtonGroup_19
3679         function ButtonGroup_19SelectionChanged(app, event)
3680
3681         end
3682
3683         % Selection changed function: ButtonGroup_22
3684         function ButtonGroup_22SelectionChanged(app, event)
3685             if app.YesButton_8.Value
3686                  app.RungeKuttaparam2EditField_2.Enable = "on";
3687                  app.RungeKuttaparam2EditFieldLabel_2.Enable = "on";
3688             elseif app.NoButton_7.Value
3689                 app.RungeKuttaparam2EditField_2.Enable = "off";
3690                 app.RungeKuttaparam2EditFieldLabel_2.Enable = "off";
3691             end
3692
3693
3694         end
3695
3696         % Button pushed function: Button_40
3697         function Button_40Pushed(app, event)
3698
3699             [files, path] = uigetfile('*.*','Select the files','MultiSelect',↙
'on');
3700               app.ImportvesselfilesTextArea.Value = string(files);
```

```
3701                 app.VesselLOC = string(path) + string(files);
3702                 app.VesselNAME = string(files); %string(path);
3703                 app.ImportvesselfilesTextArea.FontColor = 'k';
3704                 app.ImportvesselfilesTextArea.BackgroundColor = 'w';
3705 
3706                 if ischar(files)
3707                     % Only one file was selected
3708                     app.VesselNumber = 1;
3709                 elseif iscell(files)
3710                     % Multiple files were selected
3711                     app.VesselNumber = numel(files);
3712                 end
3713 
3714 
3715         end
3716 
3717         % Value changed function: FileEditField
3718         function FileEditFieldValueChanged(app, event)
3719               [file,path] = uigetfile('*.*');
3720               app.FileEditField.Value = string(file);
3721               app.FileEditField.FontColor = 'k';
3722               app.FileEditField.BackgroundColor = 'w';
3723               app.HardbottomLOC = string(path) + string(file);
3724               app.HardbottomtNAME = string(file);
3725         end
3726 
3727         % Value changed function: DirectCheckBox
3728         function DirectCheckBoxValueChanged(app, event)
3729 if app.DirectCheckBox.Value
3730     app.DecayTypeDropDown.Visible = "on";
3731     app.DecayTypeDropDownLabel.Visible = "on";
3732 else
3733     app.DecayTypeDropDown.Visible = "off";
3734     app.DecayTypeDropDownLabel.Visible = "off";
3735 end
3736 
3737 
3738 if app.DirectCheckBox.Value || app.FrictionCheckBox.Value || app.↙
DiffusionCheckBox.Value
3739     %if there are other buttons checked in Sponge Boundary, the N,E,W,S↙
textboxes are not disabled
3740     app.WestEditField.Enable = "on";
3741     app.SouthEditField.Enable = "on";
3742     app.EastEditField.Enable = "on";
3743     app.NorthEditField.Enable = "on";
3744 else
3745     %Include disable the N,E,W,S textboxes
3746     app.DecayTypeDropDown.Visible = "off";
3747     app.DecayTypeDropDownLabel.Visible = "off";
3748     app.WestEditField.Enable = "off";
3749     app.SouthEditField.Enable = "off";
3750     app.EastEditField.Enable = "off";
3751     app.NorthEditField.Enable = "off";
3752 end
3753 
3754 
3755         end
3756 
```

```
3757         % Value changed function: FrictionCheckBox
3758         function FrictionCheckBoxValueChanged(app, event)
3759             if app.FrictionCheckBox.Value
3760                 app.MaximumCdEditFieldLabel.Visible = "on";
3761                 app.MaximumCdEditField.Visible = "on";
3762             else
3763                 app.MaximumCdEditField.Visible = "off";
3764                 app.MaximumCdEditFieldLabel.Visible = "off";
3765             end
3766 
3767             if app.DirectCheckBox.Value || app.FrictionCheckBox.Value || app.↙
DiffusionCheckBox.Value
3768                 %if there are other buttons checked in Sponge Boundary, the N,↙
E,W,S textboxes are not disabled
3769                 app.WestEditField.Enable = "on";
3770                 app.SouthEditField.Enable = "on";
3771                 app.EastEditField.Enable = "on";
3772                 app.NorthEditField.Enable = "on";
3773             else
3774                 %Include disable the N,E,W,S textboxes
3775                 app.DecayTypeDropDown.Visible = "off";
3776                 app.DecayTypeDropDownLabel.Visible = "off";
3777                 app.WestEditField.Enable = "off";
3778                 app.SouthEditField.Enable = "off";
3779                 app.EastEditField.Enable = "off";
3780                 app.NorthEditField.Enable = "off";
3781             end
3782 
3783         end
3784 
3785         % Value changed function: DiffusionCheckBox
3786         function DiffusionCheckBoxValueChanged(app, event)
3787             if app.DiffusionCheckBox.Value
3788                 app.MaximumCspEditField.Visible = "on";
3789                 app.MaximumCspEditFieldLabel.Visible = "on";
3790             else
3791                 app.MaximumCspEditFieldLabel.Visible = "off";
3792                 app.MaximumCspEditField.Visible = "off";
3793             end
3794 
3795             if app.DirectCheckBox.Value || app.FrictionCheckBox.Value || app.↙
DiffusionCheckBox.Value
3796                 %if there are other buttons checked in Sponge Boundary, the N,↙
E,W,S textboxes are not disabled
3797                 app.WestEditField.Enable = "on";
3798                 app.SouthEditField.Enable = "on";
3799                 app.EastEditField.Enable = "on";
3800                 app.NorthEditField.Enable = "on";
3801             else
3802                 %Include disable the N,E,W,S textboxes
3803                 app.DecayTypeDropDown.Visible = "off";
3804                 app.DecayTypeDropDownLabel.Visible = "off";
3805                 app.WestEditField.Enable = "off";
3806                 app.SouthEditField.Enable = "off";
3807                 app.EastEditField.Enable = "off";
3808                 app.NorthEditField.Enable = "off";
3809             end
3810 
```

```
3811         end
3812
3813         % Callback function
3814         function UseBreakingIndexCheckBoxValueChanged(app, event)
3815
3816             if app.UseBreakingIndexCheckBox.Value
3817                 app.C1EditField.Visible = "on";
3818                 app.C1EditFieldLabel.Visible = "on";
3819                 app.C2EditField.Visible ="on";
3820                 app.C2EditFieldLabel.Visible = "on";
3821                 app.SchemeDropDown.Visible = "on";
3822                 app.SchemeDropDownLabel.Visible = "on";
3823             else
3824                 app.C1EditField.Visible = "off";
3825                 app.C1EditFieldLabel.Visible = "off";
3826                 app.C2EditField.Visible ="off";
3827                 app.C2EditFieldLabel.Visible = "off";
3828                 app.SchemeDropDown.Visible = "off";
3829                 app.SchemeDropDownLabel.Visible = "off";
3830             end
3831         end
3832
3833         % Callback function
3834         function UseSmagorinskymixingCheckBoxValueChanged(app, event)
3835             if app.UseSmagorinskyCheckBox.Value
3836                 app.TimeintervalsecEditField.Visible = "on";
3837                 app.TimeintervalsecEditFieldLabel.Visible = "on";
3838                 app.SteadyTimeEditField.Visible = "on";
3839                 app.SteadyTimeEditFieldLabel.Visible = "on";
3840             else
3841                 app.TimeintervalsecEditField.Visible = "off";
3842                 app.TimeintervalsecEditFieldLabel.Visible = "off";
3843                 app.SteadyTimeEditField.Visible = "off";
3844                 app.SteadyTimeEditFieldLabel.Visible = "off";
3845             end
3846         end
3847
3848         % Value changed function: AddobstaclesCheckBox
3849         function AddobstaclesCheckBoxValueChanged(app, event)
3850           if app.AddobstaclesCheckBox.Value
3851                 app.ObstacleFilename.Visible ="on";
3852                 app.ObstacleLoadFile.Visible = "on";
3853           else
3854                 app.ObstacleFilename.Visible ="off";
3855                 app.ObstacleLoadFile.Visible = "off";
3856           end
3857
3858
3859         end
3860
3861         % Value changed function: CreatenestedgridboundaryCheckBox
3862         function CreatenestedgridboundaryCheckBoxValueChanged(app, event)
3863             %Define the fields to be enabled/disabled under the nested grid ↙
option
3864             fieldsToToggle = {
3865                 app.SpacingEditFieldLabel, app.SpacingEditField, app. ↙
WestEditField_2, ...
3866                 app.SouthEditField_2, app.EastEditField_2, app. ↙
```

```
NorthEditField_2, app.BoundarylimitsButton
                };

            if app.CreatenestedgridboundaryCheckBox.Value
                % Disable the items under the 'Import gauge list' option
                app.ImportgaugelistCheckBox.Value = false;
                app.LoadFileEditField.Enable = "off";
                app.Button_7.Enable = "off";

                % Enable fields under the nested grid option
                for i = 1:length(fieldsToToggle)
                    fieldsToToggle{i}.Enable = "on";
                end
            else
                % Set font color to gray
                grayColor = '0.65,0.65,0.65';
                app.SpacingEditField.FontColor = grayColor;
                app.WestEditField_2.FontColor = grayColor;
                app.SouthEditField_2.FontColor = grayColor;
                app.EastEditField_2.FontColor = grayColor;
                app.NorthEditField_2.FontColor = grayColor;

                %Disable fields under the nested grid option
                for i = 1:length(fieldsToToggle)
                    fieldsToToggle{i}.Enable = "off";
                end
            end

        end

        % Value changing function: DepthFileEditField
        function DepthFileEditFieldValueChanging(app, event)


        end

        % Value changed function: ImportgaugelistCheckBox
        function ImportgaugelistCheckBoxValueChanged(app, event)
            app.WestEditField_2.Enable = "off";
            app.SouthEditField_2.Enable = "off";
            app.EastEditField_2.Enable = "off";
            app.NorthEditField_2.Enable = "off";
            app.BoundarylimitsButton.Enable = "off";

            if app.ImportgaugelistCheckBox.Value
                app.CreatenestedgridboundaryCheckBox.Enable = "on";
                app.LoadFileEditField.Enable = "on";
                app.Button_7.Enable = "on";
                app.CreatenestedgridboundaryCheckBox.Value = false;
                app.SpacingEditFieldLabel.Enable = "off";
                app.SpacingEditField.Enable = "off";
                app.SpacingEditFieldLabel.Enable = 'off';

            else
                app.LoadFileEditField.Enable = "off";
                app.Button_7.Enable = "off";  %To upload gauge data
                app.SpacingEditFieldLabel.Enable = "off";
                app.SpacingEditField.Enable = "off";
```

```
3924                 app.SpacingEditFieldLabel.Enable = "off";
3925
3926
3927             end
3928         end
3929
3930         % Value changed function: LoadFileEditField
3931         function LoadFileEditFieldValueChanged(app, event)
3932
3933
3934         end
3935
3936         % Value changing function: LoadFileEditField
3937         function LoadFileEditFieldValueChanging(app, event)
3938
3939
3940         end
3941
3942         % Selection change function: TabGroup2
3943         function TabGroup2SelectionChanged(app, event)
3944             selectedTab = app.TabGroup2.SelectedTab;
3945
3946         end
3947
3948         % Value changed function: UseSmagorinskyCheckBox
3949         function UseSmagorinskyCheckBoxValueChanged2(app, event)
3950            if app.UseSmagorinskyCheckBox.Value
3951            app.TimeintervalsecEditFieldLabel.Visible = "on";
3952            app.TimeintervalsecEditField.Visible = "on";
3953            app.SteadyTimeEditFieldLabel.Visible = "on";
3954            app.SteadyTimeEditField.Visible = "on";
3955         else
3956            app.TimeintervalsecEditFieldLabel.Visible = "off";
3957            app.TimeintervalsecEditField.Visible = "off";
3958            app.SteadyTimeEditFieldLabel.Visible = "off";
3959            app.SteadyTimeEditField.Visible = "off";
3960            end
3961         end
3962
3963         % Value changed function: SavefilestoEditField
3964         function SavefilestoEditFieldValueChanged(app, event)
3965
3966
3967         end
3968
3969         % Button pushed function: Button_41
3970         function Button_41Pushed(app, event)
3971                workingfolder = uigetdir;
3972                workingfolder = fullfile(workingfolder,' ');
3973                app.Mpif90EditField.Value = deblank(string(workingfolder));
3974                app.Mpif90EditField.FontColor = 'k';
3975         end
3976
3977         % Callback function
3978         function Button_42Pushed(app, event)
3979             workingfolder = uigetdir;
3980             workingfolder = fullfile(workingfolder,' ');
3981             app.MakefileEditField.Value = deblank(string(workingfolder));
```

```
3982            app.MakefileEditField.FontColor = 'k';
3983            app.MakefileEditField.BackgroundColor = 'w';
3984        end
3985
3986        % Callback function
3987        function IncludefunwaveexeinthegenerationCheckBoxValueChanged(app,
event)
3988
3989        end
3990
3991        % Callback function
3992        function MakefileEditFieldValueChanging(app, event)
3993            app.MakefileEditField.FontColor = 'k';
3994            app.MakefileEditField.BackgroundColor = 'w';
3995        end
3996
3997        % Value changed function: Mpif90EditField
3998        function Mpif90EditFieldValueChanged(app, event)
3999
4000
4001        end
4002
4003        % Value changing function: Mpif90EditField
4004        function Mpif90EditFieldValueChanging(app, event)
4005            app.Mpif90EditField.FontColor = 'k';
4006            app.Mpif90EditField.BackgroundColor = 'w';
4007        end
4008
4009        % Button pushed function: StartsimulationButton
4010        function StartsimulationButtonPushed(app, event)
4011            %Show the To disable the button whilst the rest of the code is
running
4012            app.TabGroup3.SelectedTab = app.LogReportTab;
4013            % app.StartsimulationButton.Enable = 'off';
4014            drawnow;
4015
4016            cd(fullfile(app.directory));
4017
4018
4019            if exist(app.ResultsfolderEditField.Value, 'dir')
4020                rmdir(app.ResultsfolderEditField.Value, 's'); % Delete the
pre-existing folder and its contents
4021            end
4022
4023            LogFolder = fullfile(app.directory, 'Log_Files');
4024
4025            % Check if the 'INPUTFOLDER' directory exists, create it if it
doesn't
4026            if ~exist(LogFolder, 'dir')
4027                mkdir(LogFolder);  % Creates the 'INPUTFOLDER' directory
4028            end
4029
4030            logFilePath = fullfile(LogFolder,'Log_Report_InputPrep.txt'); %
Define the log file path correctly
4031            currentLogContent = fileread(logFilePath); % Read the current
content of the log file
4032
4033            app.Preview_LogReport.Value = fileread(logFilePath);
```

```
4034            drawnow;
4035
4036            % Ensure the current directory is set to where funwave.exe resides
4037            cd(fullfile(string(app.directory)));
4038
4039
4040
4041            %In reclicking the button, delete existing Makefile and funwave.↙
exe
4042            sourceFileExe = fullfile(app.directory, "funwave");
4043            if exist(sourceFileExe,'file')
4044                delete(sourceFileExe);
4045            end
4046
4047
4048            % Determine the MPIF90 path based on user input or system defaults
4049            if isempty(app.Mpif90EditField.Value) || strcmp(app.↙
Mpif90EditField.Value, '0') %|| ~exist(fullfile(app.Mpif90EditField.Value),"dir")
4050                % Set the default MPIF90 path based on the operating system
4051                if ismac
4052                    mpiPath = '/opt/homebrew/bin/';  % macOS default path
4053                elseif ispc
4054                    mpiPath = 'C:\Program Files (x86)\MPICH2\bin\';  % Windows↙
default path
4055                else
4056                    error('Unsupported OS');  % Handle other operating systems↙
or configurations
4057                end
4058                app.Mpif90EditField.Value = fullfile(mpiPath);  % Update the↙
edit field with the default path
4059            else
4060                mpiPath = app.Mpif90EditField.Value;  % Use the user-provided↙
MPIF90 path
4061            end
4062
4063
4064            if exist(fullfile(app.Mpif90EditField.Value),"dir") == 7 %Check if↙
the mpif90 folder exists
4065                % Update the system PATH environment variable to include the↙
MPI compiler path
4066                setenv('PATH', [getenv('PATH') ':' fileparts(mpiPath)]);
4067                app.Mpif90EditField.FontColor = 'k';
4068
4069                if exist(fullfile(app.MakefileEditField.Value),"dir") == 7 %↙
Check if the Makefile Source folder exists
4070                    cd(fullfile(app.MakefileEditField.Value));
4071
4072                    % Check if the "Make Clean Completed" message is already↙
in the log file
4073                    makeCleanMessage = '> Status: Make Clean...';
4074                    if ~contains(fileread(logFilePath), makeCleanMessage)
4075                        % Append the message only if it's not already there
4076                        fileId = fopen(logFilePath, 'a');
4077                        fprintf(fileId, '%s', makeCleanMessage);
4078                        drawnow
4079                    end
4080
4081                    system('make clean');  % Clean up previous files created
```

```
4082
4083                    completedMessage = '...Completed!';
4084                    if ~contains(logFilePath, completedMessage)
4085                        fileId = fopen(logFilePath, 'a');
4086                        fprintf(fileId, '%s\n\n', completedMessage);
4087                        fclose(fileId);
4088                        app.Preview_LogReport.Value = fileread(logFilePath); %↙
Update the log report preview panel
4089                        drawnow;  % Update GUI
4090                    end
4091
4092                else
4093                    fprintf(fileId, '%-30s%-s\n','MAKEFILE:', ' Invalid path.↙
Missing .o, .F, and .f90 files');
4094                    fprintf(fileId, '%s\n', '');
4095                    app.Preview_LogReport.Value = fileread(logFilePath);
4096                    drawnow
4097                    %
4098                    % app.MakefileEditField.Value = "INVALID PATH";
4099                    % app.MakefileEditField.FontColor = 'w';
4100                    % app.MakefileEditField.BackgroundColor = 'r';
4101                end
4102
4103                %--- Execute Makefile ---
4104                generatingFunwaveMessage = '> Status: Generating the funwave.↙
exe... ';
4105                if ~contains(logFilePath, generatingFunwaveMessage)
4106                    fileId = fopen(logFilePath, 'a');
4107                    fprintf(fileId, '%s', generatingFunwaveMessage);
4108                    fclose(fileId);
4109                    app.Preview_LogReport.Value = fileread(logFilePath);
4110                    drawnow;
4111                end
4112
4113                [status, cmdout] = system('make'); % Execute 'make' command
4114
4115                if status ~= 0 %if make execution fails
4116                    if ~contains(logFilePath, 'Failed.')
4117                        fileId = fopen(logFilePath, 'a');
4118                        fprintf(fileId, '%s\n\n', 'Failed.');
4119                        app.Preview_LogReport.Value = fileread(logFilePath);
4120                        fclose(fileId);
4121                        drawnow
4122                    end
4123
4124                    makeErrorMessaage = cmdout;
4125                    if ~contains(logFilePath, makeErrorMessaage) %show the↙
error message from the terminal
4126                        fileId = fopen(logFilePath, 'a');
4127                        fprintf(fileId, '%s\n', makeErrorMessaage);
4128                        fclose(fileId);
4129                        app.Preview_LogReport.Value = fileread(logFilePath);
4130                        drawnow;
4131                    end
4132
4133                else %if successful, copy the funwave executable file to the↙
specified output directory
4134                    sourceFile = fullfile(app.MakefileEditField.Value,↙
```

```
"funwave");
4135                        if exist(sourceFile,'file')
4136                            destinationFile = fullfile(app.directory, "funwave");
4137                            copyfile(sourceFile, destinationFile);
4138                            delete(sourceFile);  % Remove the source file after↙
copying
4139
4140                            %Append "Completed" message for "make" execution
4141                            if ~contains(logFilePath, 'Completed!')
4142                                fileId = fopen(logFilePath, 'a');
4143                                fprintf(fileId, '%s\n\n', 'Completed!');
4144                                app.Preview_LogReport.Value = fileread↙
(logFilePath);
4145                                fclose(fileId);
4146                                drawnow
4147                            end
4148                            drawnow
4149
4150                            MakefileMessage5 = '> Status: Starting the↙
simulation...';
4151                            if ~contains(fileread(logFilePath), MakefileMessage5)↙
%Successful status printed
4152                                % Append the message only if it's not already↙
there
4153                                fileId = fopen(logFilePath, 'a');
4154                                fprintf(fileId, '%s\n\n', MakefileMessage5);
4155                                app.Preview_LogReport.Value =  fileread↙
(logFilePath);
4156
4157                                drawnow
4158                            end
4159                        else
4160                            if ~contains(logFilePath, 'Failed.')
4161                                fileId = fopen(logFilePath, 'a');
4162                                fprintf(fileId, '%s\n\n', 'Failed.');
4163                                app.Preview_LogReport.Value = fileread↙
(logFilePath);
4164                                fclose(fileId);
4165                                drawnow
4166                            end
4167                        end
4168                    end
4169                else
4170                    % app.Mpif90EditField.Value = "INVALID PATH";
4171                    % app.Mpif90EditField.FontColor = 'w';
4172                    % app.Mpif90EditField.BackgroundColor = 'r';
4173                    fileId = fopen(logFilePath, 'a');
4174                    fprintf(fileId, [
4175                        '%-30s%-s\n' ...
4176                        '%-30s%s\n\n' ...
4177                        '%-30s%s\n' ...
4178                        '%-30s%s\n' ...
4179                        '%-40s%s\n\n' ...
4180                        '%-30s%s\n' ...
4181                        '%-40s%s\n\n' ...
4182                        '%-30s%s\n\n'
4183                        ], ...
4184                        'MPIF90:', 'Invalid MPI path - ''mpif90'' not found', ...
```

```
4185                    '', 'Failed to execute make', ...
4186                    '', 'In your computer''s terminal, type:', ...
4187                    '', '  *Mac OS: "which mpif90"', ...
4188                    '', '     Default: /opt/homebrew/bin/', ...
4189                    '', '  *Windows: "where mpif90"', ...
4190                    '', '     Default: C:\Program Files (x86)\MPICH2\bin', ...
4191                    '', 'Copy that path into the Mpif90 Folder textbox');
4192
4193                app.Preview_LogReport.Value = fileread(logFilePath);
4194                fclose(fileId);
4195                drawnow
4196            end
4197
4198
4199
4200            cd(fullfile(app.directory));
4201
4202
4203            % Append the start message only if it's not already in the log
4204            startMessage = sprintf('> Status: Starting the tsunami↙
simulations...\n\n See the detailed progress in INPUT_FILES/LOG.txt');
4205            app.Preview_LogReport.Value = fileread(logFilePath);
4206            drawnow
4207            %
4208            if ~contains(logFilePath, startMessage)
4209                fileId = fopen(logFilePath, 'a');
4210                fprintf(fileId, '%s\n\n', startMessage);
4211                fclose(fileId); % Close the file after appending the start↙
message
4212            end
4213
4214            commandStr = ['mpirun -np ', num2str(app.ProcessorX.Value * app.↙
ProcessorY.Value), ' ./funwave'];
4215            [status, cmdout] = system(commandStr);
4216
4217            % Process cmdout to remove repeated lines and ensure unique↙
logging
4218            cmdoutLines = splitlines(string(cmdout));
4219            [uniqueLines, ~] = unique(cmdoutLines, 'stable');
4220            uniqueCmdout = join(uniqueLines, newline);
4221
4222            fileId = fopen(logFilePath, 'a'); % Reopen log file to append the↙
execution status and cmdout
4223            if status == 0
4224                runCompletedMessage = 'FUNWAVE.EXE: Run Completed';
4225                if ~contains(logFilePath, runCompletedMessage)
4226                    fprintf(fileId, '%-30s%-s\n\n', runCompletedMessage);
4227                end
4228                for line = uniqueLines'
4229                    if line ~= ""
4230                        fprintf(fileId, '   %-28s\n', line);
4231                    end
4232                end
4233            else
4234                executionFailedMessage = 'FUNWAVE.EXE: Execution failed';
4235                if ~contains(currentLogContent, executionFailedMessage)
4236                    fprintf(fileId, '%-30s%-s\n', executionFailedMessage);
4237                end
```

```
4238                 for line = uniqueLines'
4239                     if line ~= ""
4240                         fprintf(fileId, '   Error: %-25s\n', line);
4241                     end
4242                 end
4243             end
4244
4245             fclose(fileId); % Close the file
4246
4247
4248             app.Preview_LogReport.Value = fileread(logFilePath); % Display the↙
updated log content
4249
4250
4251
4252         end
4253
4254         % Button pushed function: CreateMakefileButton
4255         function CreateMakefileButtonPushed(app, event)
4256             %Disable the simulation button
4257             app.StartsimulationButton.Enable = 'off';
4258             drawnow
4259
4260             %Update the input.txt file
4261             GenerateinputtxtButtonPushed(app, event);
4262
4263             %Switch tab of the Preview Panel to the Log Report Tab
4264             app.TabGroup3.SelectedTab = app.LogReportTab;
4265
4266             %Locate the Log Report file
4267             cd(fullfile(app.directory));
4268             LogFolder = fullfile(app.directory, 'Log_Files');
4269
4270             %If'INPUT_FILES' directory does not exist, create the folder
4271             if ~exist(LogFolder, 'dir')
4272                 mkdir(LogFolder);
4273             end
4274
4275             %If the log text file does not exist, create the text file
4276             logFilePath = fullfile(LogFolder,'Log_Report_InputPrep.txt');
4277             if ~exist(logFilePath,'file')
4278                 fileId=fopen(logFilePath, 'w');
4279             else
4280                 %Update the log report
4281                 MakefileMessage00 = '******* MAKEFILE SECTION *******'; %↙
Check if the "Creating Makefile completed" message is already in the log file
4282                 if ~contains(fileread(logFilePath), MakefileMessage00)
4283                     % Append the message only if it's not already there
4284                     fileId = fopen(logFilePath, 'a');
4285                     fprintf(fileId, '%s\n\n', MakefileMessage00);
4286                     fprintf('');
4287                     app.Preview_LogReport.Value =  fileread(logFilePath);
4288                     drawnow
4289                 end
4290             end
4291
4292
4293
```

```
4294            %Remove pre-existing progress status under the Makefile section
4295            %Read the content of the log report
4296            fileContent = fileread(logFilePath);
4297
4298            % Find the index of 'Makefile Section' header
4299            startIndex = strfind(fileContent, MakefileMessage00);
4300
4301            %If there are pre-existing data in this section, overwrite them
4302            if ~isempty(startIndex)
4303                %Find the end of the line of the header
4304                endOfLineIndex = strfind(fileContent(startIndex:end),↙
newline);
4305                endIndex = startIndex + endOfLineIndex(1) - 1;
4306
4307                %Trim the fileContent up to endIndex
4308                trimmedContent = fileContent(1:endIndex);
4309
4310                %Write the trimmed content back to the file
4311                fileId = fopen(logFilePath, 'w');
4312                fprintf(fileId, '%s', trimmedContent);
4313                fclose(fileId);
4314
4315                %Update app.Preview_LogReport.Value with the updated content
4316                app.Preview_LogReport.Value = trimmedContent;
4317            else
4318                fprintf('MakefileMessage00 not found in %s\n', logFilePath);
4319            end
4320
4321            %Update the log reporrt
4322            app.Preview_LogReport.Value = fileread(logFilePath);
4323
4324
4325            %↙
-------------------------------------------------------------------
4326            %------TEMPLATE OF THE MAKEFILE from FUNWAVE-TVD-version_3.↙
6/src------
4327            %↙
-------------------------------------------------------------------
4328            makefile_template = {
4329                '#-----------BEGIN MAKEFILE--------------------------'
4330                'FUNWAVE_DIR = .'
4331                'WORK_DIR    = .'
4332                'COMPILER    = #COMPILER_OPTION#'
4333                'PARALLEL    = #PARALLEL_OPTION#'
4334                'EXEC        = funwave'
4335                'PRECISION   = #PRECISION_OPTION#'
4336                ''
4337                '#-----------DEFINE FLAGS----------------------------'
4338                '#         uncomment to choose the model'
4339                '#FLAG_1  = -DCOUPLING'
4340                '#FLAG_2  = -DZALPHA'
4341                '#FLAG_3  = -DVESSEL'
4342                '#FLAG_4  = -DMETEO'
4343                '#FLAG_5  = -DWIND'
4344                '#FLAG_6  = -DSEDIMENT'
4345                '#FLAG_7  = -DCHECK_MASS_CONSERVATION'
4346                '#FLAG_8  = -DTMP'
4347                '#FLAG_9 = -DTRACKING'
```

```
4348                 '#FLAG_10 = -DDEEP_DRAFT_VESSEL'
4349                 '#FLAG_11 = -DMIXING'
4350                 '#----------------uncommon options---------------------'
4351                 'DEF_FC      = mpif90'
4352                 'DEF_FC_FLAG = '
4353                 'SPHERICAL   = #COORDINATE_OPTION#'
4354                 'MPI         = openmpi'
4355                 'DEBUG       = true'
4356                 'INCS        = $(IOINCS) $(GOTMINCS)'
4357                 'LIBS        = $(PV3LIB) $(CLIB)  $(PARLIB) $(IOLIBS)↙
$(MPILIB) $(GOTMLIB)'
4358                 'CLIB        ='
4359                 'MDEPFLAGS   = --cpp --fext=f90 --file=-'
4360                 'RANLIB      = ranlib'
4361                 ''
4362                 '#----------include the essential makefiles------------'
4363                 '#Copied file from $(FUNWAVE_DIR)/..↙
/GNUMake/Essential/Make_Essential'
4364                 '#--------------------Use shell-----------------------'
4365                 'SHELL    = /bin/sh'
4366                 ''
4367                 '#-----Eliminate the leading and ending white space----'
4368                 'FUNWAVE_DIR := $(strip $(FUNWAVE_DIR))'
4369                 'WORK_DIR    := $(strip $(WORK_DIR))'
4370                 'COMPILER    := $(strip $(COMPILER))'
4371                 'PARALLEL    := $(strip $(PARALLEL))'
4372                 'PRECISION   := $(strip $(PRECISION))'
4373                 'EXEC        := $(strip $(EXEC))'
4374
4375                 'DEF_FC      := $(strip $(DEF_FC))'
4376                 'DEF_FC_FLAG := $(strip $(DEF_FC_FLAG))'
4377                 'MPI         := $(strip $(MPI))'
4378                 'SPHERICAL   := $(strip $(SPHERICAL))'
4379                 'DEBUG       := $(strip $(DEBUG))'
4380                 ''
4381                 '#---------------C Pre-processor----------------------'
4382                 '#  note: CPP is not C++'
4383                 'CPP       = /usr/bin/cpp'
4384                 'CPPFLAGS = -P -traditional'
4385                 ''
4386                 '#---------------Fortran Compiler--------------------'
4387                 '# if $(DEF_FC) is not empty'
4388                 'ifneq ($(DEF_FC),$(filter $(DEF_FC), ''''))'
4389                 '   FC = $(DEF_FC)'
4390                 '# if $(DEF_FC) is empty'
4391                 'else'
4392                 '   # intel'
4393                 '   ifeq ($(COMPILER),$(filter $(COMPILER), intel))'
4394                 '      USE_MOD = -module $(MOD_DIR)'
4395                 '      ifeq ($(PARALLEL),$(filter $(PARALLEL), true))'
4396                 '         ifeq ($(MPI),$(filter $(MPI), intelmpi))'
4397                 '            FC = mpiifort'
4398                 '         else'
4399                 '            FC = mpif90'
4400                 '         endif'
4401                 '      else ifeq ($(PARALLEL),$(filter $(PARALLEL), false))'
4402                 '         FC = ifort'
4403                 '      endif'
```

```
4404                 '   # gnu'
4405                 '   else ifeq ($(COMPILER),$(filter $(COMPILER), gnu))'
4406                 '      USE_MOD = -J $(MOD_DIR)'
4407                 '      ifeq ($(PARALLEL),$(filter $(PARALLEL), true))'
4408                 '         FC = mpif90'
4409                 '      else ifeq ($(PARALLEL),$(filter $(PARALLEL), false))'
4410                 '         FC = gfortran'
4411                 '      endif'
4412                 '   # pgi'
4413                 '   else ifeq ($(COMPILER),$(filter $(COMPILER), pgi))'
4414                 '      USE_MOD = -module $(MOD_DIR)'
4415                 '      ifeq ($(PARALLEL),$(filter $(PARALLEL), true))'
4416                 '         FC = mpif90'
4417                 '      else ifeq ($(PARALLEL),$(filter $(PARALLEL), false))'
4418                 '         FC = pgfortran'
4419                 '      endif'
4420                 '   # onyx hpc'
4421                 '   else ifeq ($(COMPILER),$(filter $(COMPILER), onyx))'
4422                 '      USE_MOD = -module $(MOD_DIR)'
4423                 '      FC      = ftn'
4424                 '      # Removing -lmpi flag if defined in CLIB'
4425                 '      ifneq (, $(findstring -lmpi, $(CLIB)))'
4426                 '         $(warning WARNING: lmpi flag detected in CLIB, Onyx↙
does not require explicit linking of MPI libary. Removing -lmpi from CLIB. )'
4427                 '         CLIB := $(filter-out -lmpi, $(CLIB))'
4428                 '      endif'
4429                 '      # Removing -Bdynamic flat if defined in CLIB'
4430                 '      ifneq (, $(findstring -Bdynamic, $(CLIB)))'
4431                 '         $(warning WARNING: Bdynamic flag detected in CLIB,↙
Onyx executables do not work correctly with dynamic linking of shared libaries.↙
Removing -Bdynamic from CLIB. )'
4432                 '         CLIB := $(filter-out -Bdynamic, $(CLIB))'
4433                 '      endif'
4434
4435
4436                 '   # koehr and gaffney'
4437                 '   else ifeq ($(COMPILER),$(filter $(COMPILER), koehr↙
gaffney))'
4438                 '      USE_MOD = -module $(MOD_DIR)'
4439                 '      FC      = ifort'
4440                 '   # invalid option'
4441                 '   else'
4442                 '      $(error Fatal ERROR: COMPILER=$(COMPILER) and DEF_FC is↙
empty, Please correct the COMPILER or customize the DEF_FC.)'
4443                 '   endif'
4444                 'endif'
4445                 ''
4446                 '#----------Debug and optimization flags---------------'
4447                 '# DEBUG=true'
4448                 'ifeq ($(DEBUG),$(filter $(DEBUG), true))'
4449                 '   ifeq ($(COMPILER),$(filter $(COMPILER), intel onyx gaffney↙
koehr))'
4450                 '      DEBFLGS  = -g -check -warn -fPIC'
4451                 '      OPT      = -O0'
4452                 '   else ifeq ($(COMPILER),$(filter $(COMPILER), gnu))'
4453                 '      DEBFLGS  = -g -fcheck=all -Wall -fPIC'
4454                 '      OPT      = -O0'
4455                 '   else ifeq ($(COMPILER),$(filter $(COMPILER), pgi))'
```

```
4456                '      DEBFLGS  = -g -fPIC'
4457                '      OPT      = -O0'
4458                '   endif'
4459                'endif'
4460                '# DEBUG=false'
4461                'ifeq ($(DEBUG),$(filter $(DEBUG), false))'
4462                '   ifeq ($(COMPILER),$(filter $(COMPILER), intel onyx gaffney↙
koehr))'
4463                '      DEBFLGS  = -fPIC'
4464                '      OPT      = -O2'
4465                '   else ifeq ($(COMPILER),$(filter $(COMPILER), gnu))'
4466                '      DEBFLGS  = -fPIC'
4467                '      OPT      = -O3'
4468                '   else ifeq ($(COMPILER),$(filter $(COMPILER), pgi))'
4469                '      DEBFLGS  = -w -fPIC'
4470                '      OPT      = -O2'
4471                '   endif'
4472                'endif'
4473                ''
4474                '#---------------Pre-processor flags------------------'
4475                'ifeq ($(PRECISION),$(filter $(PRECISION), double))'
4476                '   FLAG_PRECISION = -DDOUBLE_PRECISION '
4477                'else ifeq ($(PRECISION),$(filter $(PRECISION), single))'
4478                '   FLAG_PRECISION = '
4479                'else'
4480                '   $(error Fatal ERROR: PRECISION=$(PRECISION), should be↙
single or double.)'
4481                'endif'
4482                ''
4483                'ifeq ($(SPHERICAL),$(filter $(SPHERICAL), true))'
4484                '   FLAG_CARTESIAN = '
4485                'else'
4486                '   FLAG_CARTESIAN = -DCARTESIAN'
4487                'endif'
4488                ''
4489                'ifeq ($(PARALLEL),$(filter $(PARALLEL), true))'
4490                '   FLAG_PARALLEL = -DPARALLEL'
4491                'endif'
4492                ''
4493                'ifeq ($(COMPILER),$(filter $(COMPILER), intel onyx koehr↙
gaffney))'
4494                '   FLAG_INTEL = -DINTEL'
4495                'endif'
4496                ''
4497                'ifeq ($(DEBUG),$(filter $(DEBUG), true))'
4498                '   FLAG_DEBUG = -DDEBUG'
4499                'endif'
4500                '#----------------FUNWAVE flags-----------------------'
4501                'FLAG_ARGS = $(FLAG_1) $(FLAG_2) \'
4502                '            $(FLAG_3) $(FLAG_4) $(FLAG_5) $(FLAG_6) \'
4503                '            $(FLAG_7) $(FLAG_8) $(FLAG_9) $(FLAG_10)  \'
4504                '            $(FLAG_11) $(FLAG_12) $(FLAG_13) $(FLAG_14) \'
4505                '            $(FLAG_15) $(FLAG_16) $(FLAG_17) $(FLAG_18) \'
4506                '            $(FLAG_19) $(FLAG_20) $(FLAG_21) $(FLAG_22) \'
4507                '            $(FLAG_23) $(FLAG_24)'
4508                'CPPARGS = $(CPPFLAGS) $(DEF_FLAGS) $(FLAG_ARGS) \'
4509                '            $(FLAG_PARALLEL) $(FLAG_INTEL) $(FLAG_DEBUG) \'
4510                '            $(FLAG_PRECISION) $(FLAG_CARTESIAN)'
```

```
4511                '' 
4512                '# remove extra whitespace between flags' 
4513                'CPPARGS := $(shell echo $(CPPARGS) | sed ''s~ \+~ ~g'')' 
4514                '' 
4515                '#--------------------EXEC name-----------------------' 
4516                'EXEC := $(WORK_DIR)/$(EXEC)' 
4517                '' 
4518                '#-----------------Library name---------------------' 
4519                'ifneq ($(DEF_FC),$(filter $(DEF_FC), ''))' 
4520                '   FCFLAGS := $(DEF_FC_FLAG) $(INCS)' 
4521                'else' 
4522                '   FCFLAGS := $(DEBFLGS) $(OPT) $(INCS)' 
4523                'endif' 
4524                '' 
4525                '# remove extra whitespace between flags' 
4526                'FCFLAGS := $(shell echo $(FCFLAGS) | sed ''s~ \+~ ~g'')' 
4527                '' 
4528                '#---------------FUNWAVE souce code-----------------' 
4529                'MODS  = mod_param.F mod_global.F mod_input.F mod_vessel.F↙
mod_bathy_correction.F \' 
4530                '        mod_meteo.F mod_parallel_field_io.F mod_sediment.F↙
mod_tide.F\' 
4531                '        mod_tracer.F' 
4532                '' 
4533                'MAIN  = main.F bc.F fluxes.F init.F io.F tridiagnal.F↙
\' 
4534                '        breaker.F derivatives.F dispersion.F etauv_solver.F↙
\' 
4535                '        sponge.F sources.F masks.F parallel.F statistics.F \' 
4536                '        wavemaker.F mixing.F nesting.F misc.F samples.F\' 
4537                '' 
4538                '#----------------Compile process--------------------' 
4539                'SRC_DIR   = $(FUNWAVE_DIR)' 
4540                'PRE_DIR   = $(WORK_DIR)/build/pre' 
4541                'OBJ_DIR   = $(WORK_DIR)/build/obj' 
4542                'MOD_DIR   = $(WORK_DIR)/build/mod' 
4543                '' 
4544                'SRCS := $(MODS)  $(MAIN)' 
4545                'SRCS := $(patsubst %,$(SRC_DIR)/%,$(SRCS))' 
4546                'PRES := $(patsubst $(SRC_DIR)/%.F,$(PRE_DIR)/%.f90,$(SRCS))' 
4547                'OBJS := $(patsubst $(PRE_DIR)/%.f90,$(OBJ_DIR)/%.o,$(PRES))' 
4548                '' 
4549                '# keep the *.f90 files in $(PRE_DIR) after make' 
4550                '# .PRECIOUS: $(PRE_DIR)/%.f90' 
4551                '' 
4552                '# pre-processors' 
4553                '$(PRE_DIR)/%.f90 : $(SRC_DIR)/%.F' 
4554                '   $(CPP) $(CPPARGS) $< $@' 
4555                '' 
4556                '# compile objective files' 
4557                '$(OBJ_DIR)/%.o : $(PRE_DIR)/%.f90' 
4558                '   $(FC) $(FCFLAGS) -c $< -o $@ $(USE_MOD)' 
4559                '' 
4560                '# link and generate the exec file' 
4561                '$(EXEC): $(OBJS)' 
4562                '   $(FC) $(FCFLAGS) -o $(EXEC) $(OBJS) $(LIBS)' 
4563                '' 
4564                '$(PRES): | $(PRE_DIR)' 
```

```
4565                '' 
4566                '$(PRE_DIR):' 
4567                '   mkdir -p $(WORK_DIR)' 
4568                '   mkdir -p $(PRE_DIR)' 
4569                '   mkdir -p $(OBJ_DIR)' 
4570                '   mkdir -p $(MOD_DIR)' 
4571                '#--------------Cleaning targets-----------------------' 
4572                'clean:' 
4573                '       /bin/rm -rf $(WORK_DIR)/build' 
4574                '' 
4575                'clobber: clean' 
4576                '       /bin/rm -f $(EXEC)' 
4577                '' 
4578                'extra-clobber: clean' 
4579                '       /bin/rm -rf $(WORK_DIR)' 
4580                ''}; 
4581  
4582  
4583            mkfile = sprintf('%s\n',makefile_template{:}); 
4584            mkfile = string(mkfile); 
4585  
4586  
4587            %-------------------------------------------------- 
4588            %------Uncommenting the Flags in the Makefile------ 
4589            %-------------------------------------------------- 
4590  
4591            %------DOUBLE PRECISION------ 
4592            if app.DoublePrecisionCheckBox.Value 
4593                mkfile = strrep(mkfile, "#PRECISION_OPTION#", "double"); 
4594            else 
4595                mkfile = strrep(mkfile, "#PRECISION_OPTION#", "single"); 
4596            end 
4597  
4598            %------PARALLEL MODE------ 
4599            if app.ParallelModeCheckBox.Value 
4600                mkfile = strrep(mkfile, "#PARALLEL_OPTION#","true"); 
4601            else 
4602                mkfile = strrep(mkfile, "#PARALLEL_OPTION#","false"); 
4603  
4604            end 
4605  
4606            %------INTEL COMPILER------ 
4607            if app.IntelCompilerCheckBox.Value 
4608                mkfile  = strrep(mkfile, "#COMPILER_OPTION#", "intel"); 
4609            else 
4610                mkfile  = strrep(mkfile, "#COMPILER_OPTION#", "gnu"); 
4611            end 
4612  
4613  
4614            %------GRID------ 
4615            if app.SphericalButton.Value 
4616                %     mkfile = strrep(mkfile,"#FLAG_2","FLAG_2");  %Ask↙
Masashi if it is always true -DZALPHA 
4617                mkfile = strrep(mkfile,"#COORDINATE_OPTION#", "true"); 
4618            elseif app.CartesianButton.Value 
4619                mkfile = strrep(mkfile,"#COORDINATE_OPTION#", "false"); 
4620            end 
4621  
```

```
4622
4623            %------TURBULENT MIXING------
4624            if app.UseSmagorinskyCheckBox.Value
4625                mkfile = strrep(mkfile,"#FLAG_12","FLAG_12");
4626            end
4627
4628
4629            %------METEOTSUNAMI------
4630            if app.MeteotsunamiButton.Value
4631                mkfile = strrep(mkfile,"#FLAG_5","FLAG_5");
4632                % Checking if any of the switches are "On" to perform the operation once
4633                if strcmp(app.WindforceSwitch.Value, "On") || strcmp(app.HollandstormmodelSwitch.Value, "On") || strcmp(app.ConstantwindfieldSwitch.Value, "On")
4634                    mkfile = strrep(mkfile, "#FLAG_6", "FLAG_6");
4635                end
4636            end
4637
4638            %------SEDIMENT------
4639            if app.SedimentTransportCheckBox.Value
4640                mkfile = strrep(mkfile,"#FLAG_7","FLAG_7");
4641            end
4642
4643            %------VESSEL------
4644            if app.VesselButton.Value
4645                mkfile = strrep(mkfile, "#FLAG_4", "FLAG_4");
4646
4647                if app.IncludeSedimentEffectintheModelCheckBox.Value
4648                    mkfile = strrep(mkfile,"#FLAG_7","FLAG_7");
4649                end
4650
4651                % Deep-draft Vessel
4652                if app.ActivateCheckBox.Value
4653                    mkfile = strrep(mkfile, "#FLAG_11", "FLAG_11");
4654                end
4655            end
4656
4657
4658            %-------------------------------------------------------------
4659            %------Create the Makefile using the Makefile template------
4660            %-------------------------------------------------------------
4661            outputFilePath3 = fullfile(app.directory, 'Makefile');
4662            outputFile3 = fopen(outputFilePath3, 'w');
4663
4664            for i = 1:length(mkfile)
4665                fprintf(outputFile3, '%s\n',mkfile{i});
4666            end
4667            fclose(outputFile3);
4668            app.Preview_Makefile.Value = string(mkfile);
4669
4670            %Update the log report -  Check if the "Creating Makefile completed" message is already in the log file
4671            MakefileMessage = '> Status: Creating the Makefile in the INPUT_FILES folder... Completed!';
4672            if ~contains(fileread(logFilePath), MakefileMessage)
4673                % Append the message only if it's not already there
4674                fileId = fopen(logFilePath, 'a');
```

```
4675                fprintf(fileId, '\n %s\n\n', MakefileMessage);
4676                app.Preview_LogReport.Value =  fileread(logFilePath);
4677                drawnow
4678            end
4679
4680            %Texts to include in the log report
4681            MakefileMessage2 = '> Status: Copying the Makefile into the Makefile source folder... Completed!';
4682            MakefileMessage3 = '> Status: Copying the Makefile into the Makefile source folder... Failed.';
4683            MakefileMessage4 = '> Invalid path for the Makefile source';
4684
4685            %-----------------------------------------------------------------
4686            %------Copy the generated Makefile into the source folder------
4687            %-----------------------------------------------------------------
4688            %If the Makefile source folder exists, copy the in-app generated Makefile to the Makefile source folder
4689            if exist(fullfile(app.MakefileEditField.Value),"dir") == 7
4690                cd(fullfile(app.MakefileEditField.Value));
4691                sourceFile2 = fullfile(app.directory, "Makefile");
4692                destinationFile2 = fullfile(app.MakefileEditField.Value);
4693                copyfile(sourceFile2, destinationFile2);
4694
4695                %Enable the MPIf90 folder & the 'Start Simulation' button
4696                app.Mpif90EditField.Enable = 'on';
4697                app.Mpif90EditFieldLabel.Enable = 'on';
4698                app.Button_41.Enable = 'on';
4699                app.StartsimulationButton.Enable = "on";
4700                drawnow
4701
4702                %Update the log report to show that the copying of file is completed
4703                if ~contains(fileread(logFilePath), MakefileMessage2)
4704                    fileId = fopen(logFilePath, 'a');
4705                    fprintf(fileId, '%s\n\n', MakefileMessage2);
4706                    app.Preview_LogReport.Value =  fileread(logFilePath);
4707                    drawnow
4708                end
4709
4710            else %Print error status in the report
4711                if ~contains(fileread(logFilePath), MakefileMessage3)
4712                    fileId = fopen(logFilePath, 'a');
4713                    fprintf(fileId, '%s\n\n', MakefileMessage3);
4714                    fprintf(fileId, '%s\n\n', MakefileMessage4);
4715                    app.Preview_LogReport.Value =  fileread(logFilePath);
4716                    drawnow
4717                end
4718
4719                app.Preview_LogReport.Value = fileread(logFilePath);
4720                app.StartsimulationButton.Enable = "off";
4721
4722                app.MakefileEditField.Value = "INVALID PATH";
4723                app.MakefileEditField.FontColor = 'w';
4724                app.MakefileEditField.BackgroundColor = 'r';
4725                drawnow
4726            end
4727
4728            %Enable the Makefile and input.txt buttons
```

```
4729            app.CreateMakefileButton.Enable = 'on';
4730            app.GenerateinputtxtButton.Enable = 'on';
4731            drawnow
4732        end
4733
4734        % Value changed function: FrictionCoefficientEditField
4735        function FrictionCoefficientEditFieldValueChanged(app, event)
4736            app.FrictionCoefficientEditField.FontColor = 'k';
4737
4738        end
4739
4740        % Value changed function: ValueEditField
4741        function ValueEditFieldValueChanged(app, event)
4742            app.ValueEditField.FontColor = 'k';
4743
4744        end
4745
4746        % Value changed function: FrictionCheckBox_2
4747        function FrictionCheckBox_2ValueChanged(app, event)
4748            if app.FrictionCheckBox_2.Value
4749                app.FrictionCoefficientEditField.Enable = 'on';
4750                app.FrictionCoefficientEditFieldLabel.Enable = 'on';
4751            else
4752                app.FrictionCoefficientEditField.Enable = 'off';
4753                app.FrictionCoefficientEditFieldLabel.Enable = 'off';
4754            end
4755        end
4756
4757        % Value changed function: ViscosityCheckBox
4758        function ViscosityCheckBoxValueChanged(app, event)
4759            if app.ViscosityCheckBox.Value
4760                app.ValueEditField.Enable = 'on';
4761                app.ValueEditFieldLabel_2.Enable = 'on';
4762            else
4763                app.ValueEditField.Enable = 'off';
4764                app.ValueEditFieldLabel_2.Enable = 'off';
4765            end
4766
4767        end
4768
4769        % Value changed function: ActivateCheckBox
4770        function ActivateCheckBoxValueChanged(app, event)
4771            if app.ActivateCheckBox.Value
4772                if app.FrictionCheckBox_2.Value
4773                    app.FrictionCoefficientEditField.Enable = 'on';
4774                    app.FrictionCoefficientEditFieldLabel.Enable = 'on';
4775                else
4776                    app.FrictionCoefficientEditField.Enable = 'off';
4777                    app.FrictionCoefficientEditFieldLabel.Enable = 'off';
4778                end
4779
4780                if app.ViscosityCheckBox.Value
4781                    app.ValueEditField.Enable = 'on';
4782                    app.ValueEditFieldLabel_2.Enable = 'on';
4783                else
4784                    app.ValueEditField.Enable = 'off';
4785                    app.ValueEditFieldLabel_2.Enable = 'off';
4786                end
```

```
4787
4788                app.SelectMethodLabel.Enable = 'on';
4789                app.MinimumClearanceEditField.Enable = 'on';
4790                app.MinimumClearanceLabel.Enable = 'on';
4791                app.ViscosityCheckBox.Enable = 'on';
4792                app.FrictionCheckBox_2.Enable = 'on';
4793                app.ShockcapturingCheckBox.Enable = 'on';
4794            else
4795                app.SelectMethodLabel.Enable = 'off';
4796                app.MinimumClearanceEditField.Enable = 'off';
4797                app.MinimumClearanceLabel.Enable = 'off';
4798                app.ValueEditField.Enable = 'off';
4799                app.ValueEditFieldLabel_2.Enable = 'off';
4800                app.FrictionCoefficientEditField.Enable = 'off';
4801                app.FrictionCoefficientEditFieldLabel.Enable = 'off';
4802                app.ViscosityCheckBox.Enable = 'off';
4803                app.FrictionCheckBox_2.Enable = 'off';
4804                app.ShockcapturingCheckBox.Enable = 'off';
4805            end
4806        end
4807
4808        % Value changed function: MinimumClearanceEditField
4809        function MinimumClearanceEditFieldValueChanged(app, event)
4810            app.MinimumClearanceEditField.FontColor = 'k';
4811
4812        end
4813
4814        % Button pushed function: GenerateinputtxtButton_2
4815        function GenerateinputtxtButton_2Pushed(app, event)
4816 %           app.GenerateinputtxtButtonPushed(app, []);
4817           GenerateinputtxtButtonPushed(app, event);
4818        end
4819
4820        % Button pushed function: Button_42
4821        function Button_42Pushed2(app, event)
4822            workingfolder = uigetdir;
4823            workingfolder = fullfile(workingfolder,' ');
4824            app.MakefileEditField.Value = deblank(string(workingfolder));
4825            app.MakefileEditField.FontColor = 'k';
4826            app.MakefileEditField.BackgroundColor = 'w';
4827        end
4828
4829        % Value changed function: WaterlevelmEditField
4830        function WaterlevelmEditFieldValueChanged(app, event)
4831            app.WaterlevelmEditField.FontColor = 'k';
4832
4833        end
4834
4835        % Callback function
4836        function GenerateinputtxtButton_3Pushed(app, event)
4837                 GenerateinputtxtButtonPushed(app, event);
4838        end
4839
4840        % Selection changed function: ButtonGroup_23
4841        function ButtonGroup_23SelectionChanged(app, event)
4842
4843            if app.ShockwavecapturingButton.Value
4844                app.C1EditField.Value = 0.65;
```

```
4845                 app.C2EditField.Visible= 'off';
4846                 app.C2EditFieldLabel.Visible = 'off';
4847                 app.C1EditField.Visible = 'off';
4848                 app.C1EditFieldLabel.Visible = 'off';
4849             elseif app.ViscositybreakingButton.Value
4850                 app.C2EditField.Visible = "on";
4851                   app.C2EditField.Enable = "on";
4852                 app.C2EditFieldLabel.Visible = "on";
4853                  app.C2EditFieldLabel.Enable = "on";
4854                 app.C1EditField.Visible = "on";
4855                 app.C1EditField.Enable = "on";
4856                 app.C1EditFieldLabel.Visible = "on";
4857                 app.C1EditFieldLabel.Enable = "on";
4858 
4859             end
4860         end
4861 
4862         % Value changed function: InvertvaluesCheckBox
4863         function InvertvaluesCheckBoxValueChanged(app, event)
4864 
4865 
4866         end
4867 
4868         % Callback function
4869         function ImportInfoButtonPushed(app, event)
4870             [bathymetryfile,path2] = uigetfile('*.tiff;*.tif');
4871             bathymetryinputdata = string(path2) + string(bathymetryfile);
4872 
4873 
4874             if contains(bathymetryinputdata,'.tif')
4875                 try
4876                     app.depthtype = 2;
4877                     app.LatitudeEditField.Enable = "on";
4878                     app.LongitudeEditField.Enable = "on";
4879                     %                 try
4880 
4881                     [A1,R2] = readgeoraster(bathymetryinputdata);
4882                     A =  size(A1);
4883 
4884                     %Extract M and N global dimenstions
4885                     app.XEditField_2.Value = A(1,2);
4886                     app.YEditField_2.Value = A(1,1);
4887                     app.XEditField_2.FontColor = 'k';
4888                     app.YEditField_2.FontColor = 'k';
4889 
4890                     %Extract lat and long of the southwest corner
4891 
4892                     app.LongitudeEditField.Value = min(R2.LongitudeLimits);
4893                     app.LatitudeEditField.Value = min(R2.LatitudeLimits);
4894                     app.LongitudeEditField.FontColor =  'k';
4895                     app.LatitudeEditField.FontColor = 'k';
4896 
4897                     %Extract grid size of the bathymetry
4898                     app.XEditField.Value = R2.CellExtentInLatitude;
4899                     app.YEditField.Value = R2.CellExtentInLongitude;
4900                     app.XEditField.FontColor = 'k';
4901                     app.YEditField.FontColor = 'k';
4902 
```

```
4903                     if R2.AngleUnits == "degrees"
4904                         app.ButtonGroup_4.SelectedObject = app.↙
SphericalButton;
4905                         app.SphericalButton.FontColor = 'k';
4906                         app.CartesianButton.FontColor = 'k';
4907                     end
4908                     app.rasterdegrees = R2;
4909 
4910                 catch
4911                     app.ButtonGroup_4.SelectedObject = app.Button_2;
4912                 end
4913             end
4914         end
4915 
4916         % Value changed function: DoublePrecisionCheckBox
4917         function DoublePrecisionCheckBoxValueChanged(app, event)
4918 
4919 
4920         end
4921 
4922         % Button pushed function: BoundarylimitsButton
4923         function BoundarylimitsButtonPushed(app, event)
4924             [bathymetryfile,path2] = uigetfile(['*.tif;*.asc']);
4925 
4926             bathymetryinputdata = string(path2) + string(bathymetryfile);
4927 
4928 
4929             if contains(bathymetryinputdata,'.tif')
4930                 %                 try
4931                 app.depthtype = 2;
4932 
4933                 [A1,R2] = readgeoraster(bathymetryinputdata);
4934                 A =  size(A1);
4935 
4936                 %Extract M and N global dimenstions
4937                 app.CouplingX = R2.RasterSize(1,2);
4938                 app.CouplingY = R2.RasterSize(1,1);
4939 
4940                 %Extract grid size of the bathymetry
4941                 cellsize = R2.CellExtentInLatitude;
4942                 app.SpacingEditField.Value = cellsize;
4943                 app.SpacingEditField.FontColor = 'k';
4944 
4945                 % Extract and round the southwest corner coordinates
4946                 rawSouth =min(R2.LatitudeLimits);
4947                 westValue = min(R2.LongitudeLimits);
4948 
4949 
4950 
4951                 % Process based on input type
4952                 if isnumeric(rawSouth) && isscalar(rawSouth)
4953                     % Case 1: Scalar numeric input
4954                     southValue = rawSouth;
4955 
4956                 elseif ischar(rawSouth) || isstring(rawSouth)
4957                     % Case 2: String input
4958                     southValue = str2double(rawSouth);
4959 
```

```
4960                 elseif isnumeric(rawSouth) && ~isscalar(rawSouth)
4961                     % Case 3: Numeric array input
4962                     southValue = double(rawSouth);
4963                 end
4964 
4965                 % Assign to SouthEditField_2 if valid and within limits
4966                 if exist('southValue', 'var') && isnumeric(southValue) &&↙
isscalar(southValue)
4967                     southLimits = app.SouthEditField_2.Limits;
4968                     if southValue >= southLimits(1) && southValue <=↙
southLimits(2)
4969                         app.SouthEditField_2.Value = southValue;
4970                     end
4971                 end
4972 
4973 % Validate and assign west value
4974 if isnumeric(westValue) && isscalar(westValue)
4975     westLimits = app.WestEditField_2.Limits;
4976     if westValue >= westLimits(1) && westValue <= westLimits(2)
4977         app.WestEditField_2.Value = westValue;
4978     end
4979 end
4980 
4981 app.EastEditField_2.Value = westValue + (cellsize*app.CouplingX);
4982 app.NorthEditField_2.Value = southValue  + (cellsize*app.CouplingY);
4983 
4984             end
4985             %     % Initialize Xmatrix and Ymatrix
4986             %     Xmatrix = 1:app.CouplingX;
4987             %     Ymatrix = 1:app.CouplingY;
4988             %
4989             %     % Set the first element to the southwest corner
4990             %     Xmatrix(Xmatrix == 1) = min(R2.LongitudeLimits);
4991             %     Ymatrix(Ymatrix == 1) = min(R2.LatitudeLimits);
4992             %
4993             %     % Calculate CoupMatrixX and CoupMatrixY
4994             %     CoupMatrixX = Xmatrix * cellsize + min(R2.LongitudeLimits);
4995             %     CoupMatrixY = Ymatrix * cellsize + min(R2.LatitudeLimits);
4996             %
4997             %     % Recalculate Xmatrix and CoupMatrixX
4998             %     Xmatrix = 1:app.CouplingX;
4999             %     Xmatrix(Xmatrix == 1) = 0;
5000             %     for i = 1:length(Xmatrix)
5001             %         if Xmatrix(i) ~= 0
5002             %             Xmatrix(i) = Xmatrix(i) * cellsize;
5003             %         end
5004             %         CoupMatrixX(i) = Xmatrix(i) + min(R2.LongitudeLimits);
5005             %     end
5006             %
5007             %     % Recalculate Ymatrix and CoupMatrixY
5008             %     Ymatrix = 1:app.CouplingY;
5009             %     Ymatrix(Ymatrix == 1) = 0;
5010             %     for i = 1:length(Ymatrix)
5011             %         if Ymatrix(i) > 1
5012             %             Ymatrix(i) = Ymatrix(i) * cellsize;
5013             %         end
5014             %         CoupMatrixY(i) = Ymatrix(i) + min(R2.LatitudeLimits);
5015             %     end
```

```
5016            %
5017            %     % Store matrices
5018            %     app.CoupX = CoupMatrixX;
5019            %     app.CoupY = CoupMatrixY;
5020            %     app.CoupMatrixON = 0; % Flag for nested grid boundary setup
5021            %
5022            %     % Validate and assign east value
5023            %     eastValue = max(CoupMatrixX(:));
5024            %     if isnumeric(eastValue) && isscalar(eastValue)
5025            %         eastLimits = app.EastEditField_2.Limits;
5026            %         if eastValue >= eastLimits(1) && eastValue <= eastLimits
(2)
5027            %             app.EastEditField_2.Value = eastValue;
5028            %         end
5029            %     end
5030            %
5031            %     % Validate and assign north value
5032            %     northValue = max(CoupMatrixY(:));
5033            %     if isnumeric(northValue) && isscalar(northValue)
5034            %         northLimits = app.NorthEditField_2.Limits;
5035            %         if northValue >= northLimits(1) && northValue <=
northLimits(2)
5036            %             app.NorthEditField_2.Value = northValue;
5037            %         end
5038            %     end
5039            %
5040                app.SouthEditField_2.FontColor = 'k';
5041                app.WestEditField_2.FontColor = 'k';
5042                app.NorthEditField_2.FontColor = 'k';
5043                app.EastEditField_2.FontColor = 'k';
5044                drawnow
5045            % end
5046            %
5047
5048
5049
5050
5051        end
5052
5053        % Value changed function: EastEditField_2
5054        function EastEditField_2ValueChanged(app, event)
5055            app.EastEditField_2.FontColor = 'k';
5056
5057        end
5058
5059        % Value changed function: NorthEditField_2
5060        function NorthEditField_2ValueChanged(app, event)
5061            app.NorthEditField_2.FontColor = 'k';
5062
5063        end
5064
5065        % Value changed function: SouthEditField_2
5066        function SouthEditField_2ValueChanged(app, event)
5067            app.SouthEditField_2.FontColor = 'k';
5068        end
5069
5070        % Value changed function: WestEditField_2
5071        function WestEditField_2ValueChanged(app, event)
```

```
5072            app.WestEditField_2.FontColor = 'k';
5073
5074        end
5075
5076        % Value changed function: SpacingEditField
5077        function SpacingEditFieldValueChanged(app, event)
5078            app.SpacingEditField.FontColor = 'k';
5079
5080        end
5081
5082        % Size changed function: Panel_32
5083        function Panel_32SizeChanged(app, event)
5084
5085
5086        end
5087
5088        % Button pushed function: GenerateinputtxtButton_4
5089        function GenerateinputtxtButton_4Pushed(app, event)
5090              GenerateinputtxtButtonPushed(app, event);
5091        end
5092
5093        % Button pushed function: Button_43
5094        function Button_43Pushed(app, event)
5095                [file,path2] = uigetfile('*.*');
5096                app.PressurefileEditField.Value = string(file);
5097                app.PressurefileEditField.FontColor = 'k';
5098                app.PressurefileEditField.BackgroundColor = 'w';
5099                app.PressureFileLOC = string(path2) + string(file);
5100                app.PressureFileNAME = string(file);
5101        end
5102
5103        % Value changed function: SedimentTransportCheckBox
5104        function SedimentTransportCheckBoxValueChanged(app, event)
5105            %Set some 'Expected Output' parameters to T or F
5106            if app.SedimentTransportCheckBox.Value
5107                app.TabGroup2.SelectedTab = app.Tab_3;
5108                app.DepthCheckBox.Value = 1;
5109                app.UVelocityCheckBox.Value = 1;
5110                app.VVelocityCheckBox.Value = 1;
5111                app.WetdrymaskCheckBox.Value = 1;
5112                app.WaveHeightCheckBox.Value = 1;
5113            else
5114                app.DepthCheckBox.Value = 0;
5115                app.UVelocityCheckBox.Value = 0;
5116                app.VVelocityCheckBox.Value = 0;
5117                app.WetdrymaskCheckBox.Value = 0;
5118                app.WaveHeightCheckBox.Value = 0;
5119            end
5120
5121
5122        end
5123
5124        % Value changed function: MinimumEditField
5125        function MinimumEditFieldValueChanged(app, event)
5126          app.MinimumEditField.FontColor = 'k';
5127
5128        end
5129
```

```
5130        % Value changed function: MaximumEditField
5131        function MaximumEditFieldValueChanged(app, event)
5132        app.MaximumEditField.FontColor = 'k';
5133        end
5134
5135        % Value changed function: PeakEditField
5136        function PeakEditFieldValueChanged(app, event)
5137            app.PeakEditField.FontColor = 'k';
5138
5139        end
5140
5141        % Value changed function: WaveHeightmEditField
5142        function WaveHeightmEditFieldValueChanged(app, event)
5143            app.WaveHeightmEditField.FontColor = 'k';
5144
5145        end
5146
5147        % Value changed function: FroudeCapEditField
5148        function FroudeCapEditFieldValueChanged(app, event)
5149          app.FroudeCapEditField.FontColor = 'k';
5150
5151        end
5152
5153        % Value changed function: ThresholdEditField
5154        function ThresholdEditFieldValueChanged(app, event)
5155           app.ThresholdEditField.FontColor = 'k';
5156
5157        end
5158    end
5159
5160    % Component initialization
5161    methods (Access = private)
5162
5163        % Create UIFigure and components
5164        function createComponents(app)
5165
5166            % Create UIFigure and hide until all components are created
5167            app.UIFigure = uifigure('Visible', 'off');
5168            app.UIFigure.AutoResizeChildren = 'off';
5169            app.UIFigure.Position = [100 100 1146 676];
5170            app.UIFigure.Name = 'MATLAB App';
5171            app.UIFigure.Resize = 'off';
5172            app.UIFigure.Scrollable = 'on';
5173
5174            % Create TabGroup3
5175            app.TabGroup3 = uitabgroup(app.UIFigure);
5176            app.TabGroup3.AutoResizeChildren = 'off';
5177            app.TabGroup3.Position = [743 128 399 547];
5178
5179            % Create LogReportTab
5180            app.LogReportTab = uitab(app.TabGroup3);
5181            app.LogReportTab.AutoResizeChildren = 'off';
5182            app.LogReportTab.Title = 'Log Report';
5183
5184            % Create Preview_LogReport
5185            app.Preview_LogReport = uitextarea(app.LogReportTab);
5186            app.Preview_LogReport.HandleVisibility = 'off';
5187            app.Preview_LogReport.Editable = 'off';
```

```
5188            app.Preview_LogReport.WordWrap = 'off';
5189            app.Preview_LogReport.Position = [1 5 397 506];
5190
5191            % Create inputtxtTab
5192            app.inputtxtTab = uitab(app.TabGroup3);
5193            app.inputtxtTab.AutoResizeChildren = 'off';
5194            app.inputtxtTab.Title = 'input.txt';
5195
5196            % Create Preview_Input
5197            app.Preview_Input = uitextarea(app.inputtxtTab);
5198            app.Preview_Input.Editable = 'off';
5199            app.Preview_Input.Position = [1 0 397 511];
5200
5201            % Create Makefile
5202            app.Makefile = uitab(app.TabGroup3);
5203            app.Makefile.AutoResizeChildren = 'off';
5204            app.Makefile.Title = 'Makefile';
5205
5206            % Create Preview_Makefile
5207            app.Preview_Makefile = uitextarea(app.Makefile);
5208            app.Preview_Makefile.Editable = 'off';
5209            app.Preview_Makefile.Position = [1 5 397 506];
5210
5211            % Create TabGroup
5212            app.TabGroup = uitabgroup(app.UIFigure);
5213            app.TabGroup.AutoResizeChildren = 'off';
5214            app.TabGroup.Position = [1 1 737 674];
5215
5216            % Create InputTab
5217            app.InputTab = uitab(app.TabGroup);
5218            app.InputTab.AutoResizeChildren = 'off';
5219            app.InputTab.SizeChangedFcn = createCallbackFcn(app,↙
@InputTabSizeChanged, true);
5220            app.InputTab.Title = 'Input';
5221            app.InputTab.BackgroundColor = [0.9412 0.9412 0.9412];
5222            app.InputTab.ForegroundColor = [0.149 0.149 0.149];
5223            app.InputTab.Scrollable = 'on';
5224
5225            % Create Panel_27
5226            app.Panel_27 = uipanel(app.InputTab);
5227            app.Panel_27.AutoResizeChildren = 'off';
5228            app.Panel_27.Position = [371 396 356 42];
5229
5230            % Create WetDrySchemeEditFieldLabel
5231            app.WetDrySchemeEditFieldLabel = uilabel(app.Panel_27);
5232            app.WetDrySchemeEditFieldLabel.Tooltip = {'Minimum Depth (m) for↙
wet-dry scheme.'; ''; 'Suggested values:'; 'Lab scale: 0.001'; 'Field scale:↙
0.01'};
5233            app.WetDrySchemeEditFieldLabel.Position = [25 7 97 22];
5234            app.WetDrySchemeEditFieldLabel.Text = 'Wet-Dry Scheme';
5235
5236            % Create WetDrySchemeEditField
5237            app.WetDrySchemeEditField = uieditfield(app.Panel_27, 'numeric');
5238            app.WetDrySchemeEditField.Limits = [0 Inf];
5239            app.WetDrySchemeEditField.ValueDisplayFormat = '%.2f';
5240            app.WetDrySchemeEditField.ValueChangedFcn = createCallbackFcn(app,↙
@WetDrySchemeEditFieldValueChanged, true);
5241            app.WetDrySchemeEditField.FontColor = [0.651 0.651 0.651];
```

```
5242            app.WetDrySchemeEditField.Tooltip = {''};
5243            app.WetDrySchemeEditField.Position = [124 10 42 16];
5244            app.WetDrySchemeEditField.Value = 10;
5245
5246            % Create BottomFrictionEditFieldLabel
5247            app.BottomFrictionEditFieldLabel = uilabel(app.Panel_27);
5248            app.BottomFrictionEditFieldLabel.HorizontalAlignment = 'right';
5249            app.BottomFrictionEditFieldLabel.FontColor = [0.149 0.149 0.149];
5250            app.BottomFrictionEditFieldLabel.Tooltip = {'Minimum Depth (m) to limit bottom friction.'; ''; 'Suggested values:'; 'Lab scale: 0.01'; 'Field scale: 0.1'};
5251            app.BottomFrictionEditFieldLabel.Position = [188 7 89 22];
5252            app.BottomFrictionEditFieldLabel.Text = 'Bottom Friction';
5253
5254            % Create BottomFrictionEditField
5255            app.BottomFrictionEditField = uieditfield(app.Panel_27, 'numeric');
5256            app.BottomFrictionEditField.Limits = [0 Inf];
5257            app.BottomFrictionEditField.ValueDisplayFormat = '%11.2f';
5258            app.BottomFrictionEditField.ValueChangedFcn = createCallbackFcn(app, @BottomFrictionEditFieldValueChanged, true);
5259            app.BottomFrictionEditField.FontColor = [0.651 0.651 0.651];
5260            app.BottomFrictionEditField.Tooltip = {''};
5261            app.BottomFrictionEditField.Position = [284 10 41 16];
5262            app.BottomFrictionEditField.Value = 0.1;
5263
5264            % Create Panel_25
5265            app.Panel_25 = uipanel(app.InputTab);
5266            app.Panel_25.AutoResizeChildren = 'off';
5267            app.Panel_25.Position = [371 522 356 116];
5268
5269            % Create ButtonGroup_3
5270            app.ButtonGroup_3 = uibuttongroup(app.Panel_25);
5271            app.ButtonGroup_3.AutoResizeChildren = 'off';
5272            app.ButtonGroup_3.SelectionChangedFcn = createCallbackFcn(app, @ButtonGroup_3SelectionChanged, true);
5273            app.ButtonGroup_3.BorderType = 'none';
5274            app.ButtonGroup_3.Position = [7 10 344 95];
5275
5276            % Create LinearShallowWaterEquationButton
5277            app.LinearShallowWaterEquationButton = uiradiobutton(app.ButtonGroup_3);
5278            app.LinearShallowWaterEquationButton.Tooltip = {'Use linear shallow water equation.'; ''; 'When Boussinesq equation is not used, the ''HIGH_ORDER'' value under ''Numerics'' in the input.txt is set from ''THIRD'' to ''SECOND'' '};
5279            app.LinearShallowWaterEquationButton.Text = ' Linear Shallow Water Equation';
5280            app.LinearShallowWaterEquationButton.Position = [27 72 191 22];
5281
5282            % Create FullyNonlinearBoussinesqEquationButton
5283            app.FullyNonlinearBoussinesqEquationButton = uiradiobutton(app.ButtonGroup_3);
5284            app.FullyNonlinearBoussinesqEquationButton.Text = ' Fully Nonlinear Boussinesq Equation';
5285            app.FullyNonlinearBoussinesqEquationButton.Position = [27 24 224 22];
5286
```

```
5287             % Create Button
5288             app.Button = uiradiobutton(app.ButtonGroup_3);
5289             app.Button.Visible = 'off';
5290             app.Button.Text = '';
5291             app.Button.Position = [160 73 25 22];
5292             app.Button.Value = true;
5293
5294             % Create NonlinearShallowWaterEquationButton
5295             app.NonlinearShallowWaterEquationButton = uiradiobutton(app.
ButtonGroup_3);
5296             app.NonlinearShallowWaterEquationButton.Tooltip = {'Use nonlinear
shallow water equation'; ''; ''; 'When Boussinesq equation is not used, the
''HIGH_ORDER'' value under ''Numerics'' in the input.txt is set from ''THIRD'' to
''SECOND'' '};
5297             app.NonlinearShallowWaterEquationButton.Text = ' Nonlinear Shallow
Water Equation';
5298             app.NonlinearShallowWaterEquationButton.Position = [27 48 209 22];
5299
5300             % Create FroudeCapEditFieldLabel
5301             app.FroudeCapEditFieldLabel = uilabel(app.ButtonGroup_3);
5302             app.FroudeCapEditFieldLabel.HorizontalAlignment = 'right';
5303             app.FroudeCapEditFieldLabel.Tooltip = {'Froude number limit for
velocity efficiency'};
5304             app.FroudeCapEditFieldLabel.Position = [41 0 69 22];
5305             app.FroudeCapEditFieldLabel.Text = 'Froude Cap';
5306
5307             % Create FroudeCapEditField
5308             app.FroudeCapEditField = uieditfield(app.ButtonGroup_3,
'numeric');
5309             app.FroudeCapEditField.Limits = [0 Inf];
5310             app.FroudeCapEditField.ValueDisplayFormat = '%8.2f';
5311             app.FroudeCapEditField.ValueChangedFcn = createCallbackFcn(app,
@FroudeCapEditFieldValueChanged, true);
5312             app.FroudeCapEditField.FontColor = [0.651 0.651 0.651];
5313             app.FroudeCapEditField.Position = [119 3 41 16];
5314             app.FroudeCapEditField.Value = 10;
5315
5316             % Create Panel_17
5317             app.Panel_17 = uipanel(app.InputTab);
5318             app.Panel_17.AutoResizeChildren = 'off';
5319             app.Panel_17.Position = [10 596 351 42];
5320
5321             % Create Panel_5
5322             app.Panel_5 = uipanel(app.Panel_17);
5323             app.Panel_5.AutoResizeChildren = 'off';
5324             app.Panel_5.BorderType = 'none';
5325             app.Panel_5.Position = [43 4 264 30];
5326
5327             % Create ProcessornumbersXLabel
5328             app.ProcessornumbersXLabel = uilabel(app.Panel_5);
5329             app.ProcessornumbersXLabel.BackgroundColor = [0.9412 0.9412
0.9412];
5330             app.ProcessornumbersXLabel.HorizontalAlignment = 'right';
5331             app.ProcessornumbersXLabel.Tooltip = {'Must be consistent with
mpirun -np n (px*py)'};
5332             app.ProcessornumbersXLabel.Position = [3 3 134 22];
5333             app.ProcessornumbersXLabel.Text = 'Processor numbers:    X';
5334
```

```
5335            % Create ProcessorX
5336            app.ProcessorX = uieditfield(app.Panel_5, 'numeric');
5337            app.ProcessorX.Limits = [0 Inf];
5338            app.ProcessorX.ValueChangedFcn = createCallbackFcn(app,
@ProcessorXValueChanged, true);
5339            app.ProcessorX.FontColor = [0.651 0.651 0.651];
5340            app.ProcessorX.Tooltip = {''};
5341            app.ProcessorX.Position = [146 6 34 16];
5342            app.ProcessorX.Value = 2;
5343
5344            % Create YLabel
5345            app.YLabel = uilabel(app.Panel_5);
5346            app.YLabel.HorizontalAlignment = 'right';
5347            app.YLabel.Tooltip = {'Must be consistent with mpirun -np n
(px*py)'};
5348            app.YLabel.Position = [180 3 35 22];
5349            app.YLabel.Text = 'Y';
5350
5351            % Create ProcessorY
5352            app.ProcessorY = uieditfield(app.Panel_5, 'numeric');
5353            app.ProcessorY.ValueChangedFcn = createCallbackFcn(app,
@ProcessorYValueChanged, true);
5354            app.ProcessorY.FontColor = [0.651 0.651 0.651];
5355            app.ProcessorY.Tooltip = {''};
5356            app.ProcessorY.Position = [223 6 32 16];
5357            app.ProcessorY.Value = 2;
5358
5359            % Create ParallelInfoLabel
5360            app.ParallelInfoLabel = uilabel(app.InputTab);
5361            app.ParallelInfoLabel.BackgroundColor = [0.9412 0.9412 0.9412];
5362            app.ParallelInfoLabel.FontSize = 13;
5363            app.ParallelInfoLabel.FontWeight = 'bold';
5364            app.ParallelInfoLabel.FontColor = [0.0314 0.3686 0.6];
5365            app.ParallelInfoLabel.Position = [21 626 89 22];
5366            app.ParallelInfoLabel.Text = '  Parallel Info  ';
5367
5368            % Create Friction_____________________________Label_2
5369            app.Friction_____________________________Label_2 = uilabel(app.
InputTab);
5370            app.Friction_____________________________Label_2.
BackgroundColor = [0.9412 0.9412 0.9412];
5371            app.Friction_____________________________Label_2.FontSize = 13;
5372            app.Friction_____________________________Label_2.FontWeight =
'bold';
5373            app.Friction_____________________________Label_2.FontColor =
[0.0314 0.3686 0.6];
5374            app.Friction_____________________________Label_2.Position =
[383 427 111 22];
5375            app.Friction_____________________________Label_2.Text = '
Minimum Depth';
5376
5377            % Create Panel_18
5378            app.Panel_18 = uipanel(app.InputTab);
5379            app.Panel_18.AutoResizeChildren = 'off';
5380            app.Panel_18.Position = [10 501 351 84];
5381
5382            % Create ButtonGroup_7
5383            app.ButtonGroup_7 = uibuttongroup(app.Panel_18);
```

```
5384            app.ButtonGroup_7.AutoResizeChildren = 'off';
5385            app.ButtonGroup_7.SelectionChangedFcn = createCallbackFcn(app, ↙
@ButtonGroup_7SelectionChanged, true);
5386            app.ButtonGroup_7.ForegroundColor = [0.149 0.149 0.149];
5387            app.ButtonGroup_7.BorderType = 'none';
5388            app.ButtonGroup_7.Position = [25 9 100 64];
5389
5390            % Create ImportDataButton
5391            app.ImportDataButton = uiradiobutton(app.ButtonGroup_7);
5392            app.ImportDataButton.Tooltip = {'Import the bathymetry.'; '';↙
'Accepted formats: .txt, .tif'; ''; ' If a .tif file is uploaded, it will be↙
automatically converted to a text file. In the conversion, the array is flipped up↙
to down as FUNWAVE reads the depth file starting from the southwest corner.'};
5393            app.ImportDataButton.Text = ' Import Data';
5394            app.ImportDataButton.Position = [5 41 89 22];
5395
5396            % Create FlatButton
5397            app.FlatButton = uiradiobutton(app.ButtonGroup_7);
5398            app.FlatButton.Tooltip = {'Create and use a flat bottom↙
bathymetry'};
5399            app.FlatButton.Text = ' Flat';
5400            app.FlatButton.Position = [5 19 45 22];
5401
5402            % Create SlopeButton
5403            app.SlopeButton = uiradiobutton(app.ButtonGroup_7);
5404            app.SlopeButton.Tooltip = {'Create and use a sloping bathymetry↙
'};
5405            app.SlopeButton.Text = ' Slope';
5406            app.SlopeButton.Position = [5 -3 56 22];
5407
5408            % Create Button4
5409            app.Button4 = uiradiobutton(app.ButtonGroup_7);
5410            app.Button4.Visible = 'off';
5411            app.Button4.Text = 'Button4';
5412            app.Button4.Position = [57 -4 27 22];
5413            app.Button4.Value = true;
5414
5415            % Create DepthFileEditField
5416            app.DepthFileEditField = uieditfield(app.Panel_18, 'text');
5417            app.DepthFileEditField.ValueChangedFcn = createCallbackFcn(app, ↙
@DepthFileEditFieldValueChanged, true);
5418            app.DepthFileEditField.ValueChangingFcn = createCallbackFcn(app, ↙
@DepthFileEditFieldValueChanging, true);
5419            app.DepthFileEditField.Editable = 'off';
5420            app.DepthFileEditField.FontSize = 11;
5421            app.DepthFileEditField.FontColor = [0 0 1];
5422            app.DepthFileEditField.Visible = 'off';
5423            app.DepthFileEditField.Position = [133 52 144 17];
5424
5425            % Create Button_8
5426            app.Button_8 = uibutton(app.Panel_18, 'push');
5427            app.Button_8.ButtonPushedFcn = createCallbackFcn(app, ↙
@Button_8Pushed, true);
5428            app.Button_8.VerticalAlignment = 'top';
5429            app.Button_8.FontSize = 10;
5430            app.Button_8.FontAngle = 'italic';
5431            app.Button_8.FontColor = [0 0.451 0.7412];
5432            app.Button_8.Visible = 'off';
```

```
5433            app.Button_8.Tooltip = {'Import the bathymetry.'; ''; 'Accepted formats: .txt, .tif'; ''; 'If a .tif file is uploaded, it will be automatically converted to a text file.'; ''; 'The dimensions and grid details are automatically extracted from the file.'; ''; 'The values in the .txt file should include decimal places.'};
5434            app.Button_8.Position = [285 51 19 18];
5435            app.Button_8.Text = '...';
5436
5437            % Create WaterlevelmEditField
5438            app.WaterlevelmEditField = uieditfield(app.Panel_18, 'numeric');
5439            app.WaterlevelmEditField.Limits = [0 Inf];
5440            app.WaterlevelmEditField.ValueDisplayFormat = '%5.2f';
5441            app.WaterlevelmEditField.ValueChangedFcn = createCallbackFcn(app, @WaterlevelmEditFieldValueChanged, true);
5442            app.WaterlevelmEditField.FontColor = [0.651 0.651 0.651];
5443            app.WaterlevelmEditField.Visible = 'off';
5444            app.WaterlevelmEditField.Tooltip = {''};
5445            app.WaterlevelmEditField.Position = [233 9 44 17];
5446
5447            % Create WaterlevelmEditFieldLabel
5448            app.WaterlevelmEditFieldLabel = uilabel(app.Panel_18);
5449            app.WaterlevelmEditFieldLabel.Visible = 'off';
5450            app.WaterlevelmEditFieldLabel.Tooltip = {'Adjust the initial water level of the input bathymetry.'; ''; 'For example, to add surge or tide effects.'};
5451            app.WaterlevelmEditFieldLabel.Position = [133 6 96 22];
5452            app.WaterlevelmEditFieldLabel.Text = 'Initial waver level';
5453
5454            % Create InvertvaluesCheckBox
5455            app.InvertvaluesCheckBox = uicheckbox(app.Panel_18);
5456            app.InvertvaluesCheckBox.ValueChangedFcn = createCallbackFcn(app, @InvertvaluesCheckBoxValueChanged, true);
5457            app.InvertvaluesCheckBox.Visible = 'off';
5458            app.InvertvaluesCheckBox.Tooltip = {'Convert bathymetry values to positive and land values to negative'};
5459            app.InvertvaluesCheckBox.Text = 'Invert values';
5460            app.InvertvaluesCheckBox.WordWrap = 'on';
5461            app.InvertvaluesCheckBox.Position = [133 28 105 22];
5462
5463            % Create DepthmEditField
5464            app.DepthmEditField = uieditfield(app.Panel_18, 'numeric');
5465            app.DepthmEditField.Limits = [0 Inf];
5466            app.DepthmEditField.ValueDisplayFormat = '%5.2f';
5467            app.DepthmEditField.ValueChangedFcn = createCallbackFcn(app, @DepthmEditFieldValueChanged, true);
5468            app.DepthmEditField.FontColor = [0.651 0.651 0.651];
5469            app.DepthmEditField.Visible = 'off';
5470            app.DepthmEditField.Tooltip = {''};
5471            app.DepthmEditField.Position = [190 31 48 17];
5472            app.DepthmEditField.Value = 100;
5473
5474            % Create DepthmEditFieldLabel
5475            app.DepthmEditFieldLabel = uilabel(app.Panel_18);
5476            app.DepthmEditFieldLabel.HorizontalAlignment = 'right';
5477            app.DepthmEditFieldLabel.Visible = 'off';
5478            app.DepthmEditFieldLabel.Tooltip = {'Unit: meters'};
5479            app.DepthmEditFieldLabel.Position = [126 29 41 22];
5480            app.DepthmEditFieldLabel.Text = ' Depth';
5481
```

```
5482            % Create StartatXEditFieldLabel
5483            app.StartatXEditFieldLabel = uilabel(app.Panel_18);
5484            app.StartatXEditFieldLabel.HorizontalAlignment = 'right';
5485            app.StartatXEditFieldLabel.Visible = 'off';
5486            app.StartatXEditFieldLabel.Tooltip = {'The location where the↙
sloping beach begins, with a zero value indicating the deeper part of the water.';↙
''; 'Unit: meters'};
5487            app.StartatXEditFieldLabel.Position = [127 6 56 22];
5488            app.StartatXEditFieldLabel.Text = 'Start at X';
5489
5490            % Create StartatXEditField
5491            app.StartatXEditField = uieditfield(app.Panel_18, 'numeric');
5492            app.StartatXEditField.ValueDisplayFormat = '%3.2f';
5493            app.StartatXEditField.ValueChangedFcn = createCallbackFcn(app,↙
@StartatXEditFieldValueChanged, true);
5494            app.StartatXEditField.FontColor = [0.651 0.651 0.651];
5495            app.StartatXEditField.Visible = 'off';
5496            app.StartatXEditField.Tooltip = {''};
5497            app.StartatXEditField.Position = [190 9 48 17];
5498
5499            % Create AngleEditFieldLabel
5500            app.AngleEditFieldLabel = uilabel(app.Panel_18);
5501            app.AngleEditFieldLabel.HorizontalAlignment = 'right';
5502            app.AngleEditFieldLabel.Visible = 'off';
5503            app.AngleEditFieldLabel.Tooltip = {'Slope angle'};
5504            app.AngleEditFieldLabel.Position = [248 8 36 22];
5505            app.AngleEditFieldLabel.Text = 'Angle';
5506
5507            % Create AngleEditField
5508            app.AngleEditField = uieditfield(app.Panel_18, 'numeric');
5509            app.AngleEditField.ValueDisplayFormat = '%3.2f';
5510            app.AngleEditField.ValueChangedFcn = createCallbackFcn(app,↙
@AngleEditFieldValueChanged, true);
5511            app.AngleEditField.FontColor = [0.651 0.651 0.651];
5512            app.AngleEditField.Visible = 'off';
5513            app.AngleEditField.Tooltip = {''};
5514            app.AngleEditField.Position = [289 11 42 16];
5515
5516            % Create Panel_19
5517            app.Panel_19 = uipanel(app.InputTab);
5518            app.Panel_19.AutoResizeChildren = 'off';
5519            app.Panel_19.Position = [10 454 351 36];
5520
5521            % Create YEditField_2Label
5522            app.YEditField_2Label = uilabel(app.Panel_19);
5523            app.YEditField_2Label.HorizontalAlignment = 'right';
5524            app.YEditField_2Label.Tooltip = {'Matrix size in Y direction'};
5525            app.YEditField_2Label.Position = [182 6 16 22];
5526            app.YEditField_2Label.Text = 'Y';
5527
5528            % Create YEditField_2
5529            app.YEditField_2 = uieditfield(app.Panel_19, 'numeric');
5530            app.YEditField_2.ValueDisplayFormat = '%.0f';
5531            app.YEditField_2.ValueChangedFcn = createCallbackFcn(app,↙
@YEditField_2ValueChanged, true);
5532            app.YEditField_2.FontColor = [0.651 0.651 0.651];
5533            app.YEditField_2.Tooltip = {''};
5534            app.YEditField_2.Position = [204 9 45 16];
```

```
5535
5536            % Create XEditField_2Label
5537            app.XEditField_2Label = uilabel(app.Panel_19);
5538            app.XEditField_2Label.HorizontalAlignment = 'right';
5539            app.XEditField_2Label.Tooltip = {'Matrix size in X direction'};
5540            app.XEditField_2Label.Position = [97 6 18 22];
5541            app.XEditField_2Label.Text = 'X ';
5542
5543            % Create XEditField_2
5544            app.XEditField_2 = uieditfield(app.Panel_19, 'numeric');
5545            app.XEditField_2.Limits = [0 Inf];
5546            app.XEditField_2.ValueDisplayFormat = '%.0f';
5547            app.XEditField_2.ValueChangedFcn = createCallbackFcn(app,↙
@XEditField_2ValueChanged, true);
5548            app.XEditField_2.FontColor = [0.651 0.651 0.651];
5549            app.XEditField_2.Tooltip = {''};
5550            app.XEditField_2.Position = [119 9 45 16];
5551
5552            % Create Panel_20
5553            app.Panel_20 = uipanel(app.InputTab);
5554            app.Panel_20.AutoResizeChildren = 'off';
5555            app.Panel_20.Position = [10 403 351 39];
5556
5557            % Create ResultsfolderEditFieldLabel
5558            app.ResultsfolderEditFieldLabel = uilabel(app.Panel_20);
5559            app.ResultsfolderEditFieldLabel.Tooltip = {'Folder where  the↙
tsunami simulations will be saved'};
5560            app.ResultsfolderEditFieldLabel.Position = [53 6 79 22];
5561            app.ResultsfolderEditFieldLabel.Text = 'Results folder';
5562
5563            % Create ResultsfolderEditField
5564            app.ResultsfolderEditField = uieditfield(app.Panel_20, 'text');
5565            app.ResultsfolderEditField.ValueChangedFcn = createCallbackFcn↙
(app, @ResultsfolderEditFieldValueChanged, true);
5566            app.ResultsfolderEditField.ValueChangingFcn = createCallbackFcn↙
(app, @ResultsfolderEditFieldValueChanging, true);
5567            app.ResultsfolderEditField.FontSize = 11;
5568            app.ResultsfolderEditField.FontColor = [0 0 1];
5569            app.ResultsfolderEditField.Tooltip = {''};
5570            app.ResultsfolderEditField.Placeholder = 'Default:↙
"Simulation_Results/"';
5571            app.ResultsfolderEditField.Position = [139 9 178 16];
5572
5573            % Create PrintLabel
5574            app.PrintLabel = uilabel(app.InputTab);
5575            app.PrintLabel.BackgroundColor = [0.9412 0.9412 0.9412];
5576            app.PrintLabel.FontSize = 13;
5577            app.PrintLabel.FontWeight = 'bold';
5578            app.PrintLabel.FontColor = [0.0314 0.3686 0.6];
5579            app.PrintLabel.Position = [21 432 42 21];
5580            app.PrintLabel.Text = '  Print';
5581
5582            % Create Panel_21
5583            app.Panel_21 = uipanel(app.InputTab);
5584            app.Panel_21.AutoResizeChildren = 'off';
5585            app.Panel_21.Position = [10 332 351 60];
5586
5587            % Create ScreenLoadingIntervalLabel
```

```
5588            app.ScreenLoadingIntervalLabel = uilabel(app.Panel_21);
5589            app.ScreenLoadingIntervalLabel.WordWrap = 'on';
5590            app.ScreenLoadingIntervalLabel.Tooltip = {'Initila time'};
5591            app.ScreenLoadingIntervalLabel.Position = [55 27 31 22];
5592            app.ScreenLoadingIntervalLabel.Text = 'Start';
5593
5594            % Create StartEditField
5595            app.StartEditField = uieditfield(app.Panel_21, 'numeric');
5596            app.StartEditField.Limits = [0 Inf];
5597            app.StartEditField.ValueDisplayFormat = '%.1f';
5598            app.StartEditField.ValueChangedFcn = createCallbackFcn(app,↙
@StartEditFieldValueChanged, true);
5599            app.StartEditField.FontColor = [0.651 0.651 0.651];
5600            app.StartEditField.Tooltip = {''};
5601            app.StartEditField.Position = [90 30 52 16];
5602
5603            % Create GaugeIntervalEditFieldLabel
5604            app.GaugeIntervalEditFieldLabel = uilabel(app.Panel_21);
5605            app.GaugeIntervalEditFieldLabel.HorizontalAlignment = 'right';
5606            app.GaugeIntervalEditFieldLabel.Tooltip = {'Time interval between↙
station recordings.'; ''; 'Use 0.0 when the stations are for the nesting data.';↙
''; 'Unit: seconds'};
5607            app.GaugeIntervalEditFieldLabel.Position = [154 3 110 28];
5608            app.GaugeIntervalEditFieldLabel.Text = 'Station Plot Interval';
5609
5610            % Create GaugeIntervalEditField
5611            app.GaugeIntervalEditField = uieditfield(app.Panel_21, 'numeric');
5612            app.GaugeIntervalEditField.Limits = [0 Inf];
5613            app.GaugeIntervalEditField.ValueDisplayFormat = '%.1f';
5614            app.GaugeIntervalEditField.ValueChangedFcn = createCallbackFcn↙
(app, @GaugeIntervalEditFieldValueChanged, true);
5615            app.GaugeIntervalEditField.FontColor = [0.651 0.651 0.651];
5616            app.GaugeIntervalEditField.Tooltip = {''};
5617            app.GaugeIntervalEditField.Position = [270 9 47 16];
5618
5619            % Create SimulationIntervalEditFieldLabel
5620            app.SimulationIntervalEditFieldLabel = uilabel(app.Panel_21);
5621            app.SimulationIntervalEditFieldLabel.HorizontalAlignment =↙
'right';
5622            app.SimulationIntervalEditFieldLabel.WordWrap = 'on';
5623            app.SimulationIntervalEditFieldLabel.Tooltip = {'Interval for↙
generating simulation output'; ''; 'Unit: seconds'};
5624            app.SimulationIntervalEditFieldLabel.Position = [154 29 110 18];
5625            app.SimulationIntervalEditFieldLabel.Text = 'Simulation Interval';
5626
5627            % Create SimulationIntervalEditField
5628            app.SimulationIntervalEditField = uieditfield(app.Panel_21,↙
'numeric');
5629            app.SimulationIntervalEditField.Limits = [0 Inf];
5630            app.SimulationIntervalEditField.ValueDisplayFormat = '%.1f';
5631            app.SimulationIntervalEditField.ValueChangedFcn =↙
createCallbackFcn(app, @SimulationIntervalEditFieldValueChanged, true);
5632            app.SimulationIntervalEditField.FontColor = [0.651 0.651 0.651];
5633            app.SimulationIntervalEditField.Tooltip = {''};
5634            app.SimulationIntervalEditField.Position = [270 30 47 16];
5635
5636            % Create TotalEditFieldLabel
5637            app.TotalEditFieldLabel = uilabel(app.Panel_21);
```

```
5638            app.TotalEditFieldLabel.WordWrap = 'on';
5639            app.TotalEditFieldLabel.Tooltip = {'Total simulation time'; ''; 'Unit: seconds'};
5640            app.TotalEditFieldLabel.Position = [55 6 34 22];
5641            app.TotalEditFieldLabel.Text = 'Total';
5642
5643            % Create TotalEditField
5644            app.TotalEditField = uieditfield(app.Panel_21, 'numeric');
5645            app.TotalEditField.Limits = [0 Inf];
5646            app.TotalEditField.ValueDisplayFormat = '%.1f';
5647            app.TotalEditField.ValueChangedFcn = createCallbackFcn(app, @TotalEditFieldValueChanged2, true);
5648            app.TotalEditField.FontColor = [0.651 0.651 0.651];
5649            app.TotalEditField.Tooltip = {''};
5650            app.TotalEditField.Position = [90 9 52 16];
5651            app.TotalEditField.Value = 3600;
5652
5653            % Create TimeLabel_2
5654            app.TimeLabel_2 = uilabel(app.InputTab);
5655            app.TimeLabel_2.BackgroundColor = [0.9412 0.9412 0.9412];
5656            app.TimeLabel_2.FontSize = 13;
5657            app.TimeLabel_2.FontWeight = 'bold';
5658            app.TimeLabel_2.FontColor = [0.0314 0.3686 0.6];
5659            app.TimeLabel_2.Position = [22 381 47 21];
5660            app.TimeLabel_2.Text = '  Time';
5661
5662            % Create Panel_22
5663            app.Panel_22 = uipanel(app.InputTab);
5664            app.Panel_22.AutoResizeChildren = 'off';
5665            app.Panel_22.Position = [10 214 351 107];
5666
5667            % Create XEditFieldLabel
5668            app.XEditFieldLabel = uilabel(app.Panel_22);
5669            app.XEditFieldLabel.HorizontalAlignment = 'right';
5670            app.XEditFieldLabel.Tooltip = {''};
5671            app.XEditFieldLabel.Position = [214 29 25 22];
5672            app.XEditFieldLabel.Text = 'X ';
5673
5674            % Create XEditField
5675            app.XEditField = uieditfield(app.Panel_22, 'numeric');
5676            app.XEditField.Limits = [0 Inf];
5677            app.XEditField.ValueDisplayFormat = '%8.5f';
5678            app.XEditField.ValueChangedFcn = createCallbackFcn(app, @XEditFieldValueChanged, true);
5679            app.XEditField.FontColor = [0.651 0.651 0.651];
5680            app.XEditField.Tooltip = {''};
5681            app.XEditField.Position = [245 33 57 16];
5682
5683            % Create YgridsizeLabel
5684            app.YgridsizeLabel = uilabel(app.Panel_22);
5685            app.YgridsizeLabel.HorizontalAlignment = 'right';
5686            app.YgridsizeLabel.Position = [211 7 25 22];
5687            app.YgridsizeLabel.Text = 'Y';
5688
5689            % Create YEditField
5690            app.YEditField = uieditfield(app.Panel_22, 'numeric');
5691            app.YEditField.Limits = [0 Inf];
5692            app.YEditField.ValueDisplayFormat = '%8.5f';
```

```
5693             app.YEditField.ValueChangedFcn = createCallbackFcn(app, ↙
@YEditFieldValueChanged, true);
5694             app.YEditField.FontColor = [0.651 0.651 0.651];
5695             app.YEditField.Tooltip = {''};
5696             app.YEditField.Position = [245 11 57 16];
5697 
5698             % Create LongitudeEditFieldLabel
5699             app.LongitudeEditFieldLabel = uilabel(app.Panel_22);
5700             app.LongitudeEditFieldLabel.WordWrap = 'on';
5701             app.LongitudeEditFieldLabel.Tooltip = {''};
5702             app.LongitudeEditFieldLabel.Position = [29 31 60 22];
5703             app.LongitudeEditFieldLabel.Text = 'Longitude';
5704 
5705             % Create LongitudeEditField
5706             app.LongitudeEditField = uieditfield(app.Panel_22, 'numeric');
5707             app.LongitudeEditField.ValueDisplayFormat = '%11.5f';
5708             app.LongitudeEditField.ValueChangedFcn = createCallbackFcn(app, ↙
@LongitudeEditFieldValueChanged, true);
5709             app.LongitudeEditField.FontColor = [0.651 0.651 0.651];
5710             app.LongitudeEditField.Enable = 'off';
5711             app.LongitudeEditField.Tooltip = {''};
5712             app.LongitudeEditField.Position = [88 34 76 16];
5713 
5714             % Create LatitudeEditFieldLabel
5715             app.LatitudeEditFieldLabel = uilabel(app.Panel_22);
5716             app.LatitudeEditFieldLabel.WordWrap = 'on';
5717             app.LatitudeEditFieldLabel.Tooltip = {''};
5718             app.LatitudeEditFieldLabel.Position = [29 6 46 28];
5719             app.LatitudeEditFieldLabel.Text = 'Latitude';
5720 
5721             % Create LatitudeEditField
5722             app.LatitudeEditField = uieditfield(app.Panel_22, 'numeric');
5723             app.LatitudeEditField.ValueDisplayFormat = '%11.5f';
5724             app.LatitudeEditField.ValueChangedFcn = createCallbackFcn(app, ↙
@LatitudeEditFieldValueChanged, true);
5725             app.LatitudeEditField.FontColor = [0.651 0.651 0.651];
5726             app.LatitudeEditField.Enable = 'off';
5727             app.LatitudeEditField.Tooltip = {''};
5728             app.LatitudeEditField.Position = [88 11 76 17];
5729 
5730             % Create ButtonGroup_4
5731             app.ButtonGroup_4 = uibuttongroup(app.Panel_22);
5732             app.ButtonGroup_4.AutoResizeChildren = 'off';
5733             app.ButtonGroup_4.SelectionChangedFcn = createCallbackFcn(app, ↙
@ButtonGroup_4SelectionChanged, true);
5734             app.ButtonGroup_4.Tooltip = {'Grid coordinate'};
5735             app.ButtonGroup_4.BorderType = 'none';
5736             app.ButtonGroup_4.Position = [82 69 187 30];
5737 
5738             % Create SphericalButton
5739             app.SphericalButton = uiradiobutton(app.ButtonGroup_4);
5740             app.SphericalButton.Tooltip = {'Unit: degrees'};
5741             app.SphericalButton.Text = ' Spherical';
5742             app.SphericalButton.Position = [11 7 76 22];
5743 
5744             % Create CartesianButton
5745             app.CartesianButton = uiradiobutton(app.ButtonGroup_4);
5746             app.CartesianButton.Tooltip = {'Unit: meters'};
```

```
5747            app.CartesianButton.Text = ' Cartesian';
5748            app.CartesianButton.FontColor = [0.149 0.149 0.149];
5749            app.CartesianButton.Position = [99 7 76 22];
5750
5751            % Create Button_2
5752            app.Button_2 = uiradiobutton(app.ButtonGroup_4);
5753            app.Button_2.Visible = 'off';
5754            app.Button_2.Text = '';
5755            app.Button_2.Position = [181 7 25 22];
5756            app.Button_2.Value = true;
5757
5758            % Create SouthwestcornerLabel
5759            app.SouthwestcornerLabel = uilabel(app.Panel_22);
5760            app.SouthwestcornerLabel.Tooltip = {'The corner coordinates of the↙
numerical grid.'; ''; 'Unit: degrees / meters'};
5761            app.SouthwestcornerLabel.Position = [28 50 104 22];
5762            app.SouthwestcornerLabel.Text = 'Southwest corner:';
5763
5764            % Create GridSizeLabel
5765            app.GridSizeLabel = uilabel(app.Panel_22);
5766            app.GridSizeLabel.Tooltip = {'Spatial resolution of the numerical↙
grid.'; ''; 'Unit: meters'};
5767            app.GridSizeLabel.Position = [227 50 58 22];
5768            app.GridSizeLabel.Text = 'Grid Size:';
5769
5770            % Create GridLabel
5771            app.GridLabel = uilabel(app.InputTab);
5772            app.GridLabel.BackgroundColor = [0.9412 0.9412 0.9412];
5773            app.GridLabel.FontSize = 13;
5774            app.GridLabel.FontWeight = 'bold';
5775            app.GridLabel.FontColor = [0.0314 0.3686 0.6];
5776            app.GridLabel.Position = [22 310 46 21];
5777            app.GridLabel.Text = '  Grid  ';
5778
5779            % Create Panel_23
5780            app.Panel_23 = uipanel(app.InputTab);
5781            app.Panel_23.AutoResizeChildren = 'off';
5782            app.Panel_23.Position = [10 59 351 144];
5783
5784            % Create LayerwidthLabel
5785            app.LayerwidthLabel = uilabel(app.Panel_23);
5786            app.LayerwidthLabel.Tooltip = {'Sponge layer thickness relative to↙
model boundary. The width should be at least half of the wavelength.'; ''; 'Unit:↙
meters'; ''; 'Example: '; 'Entering 10,000 in the ''North'' textbox extends the↙
sponge layer 10,000 meters southward from the North boundary.'};
5787            app.LayerwidthLabel.Position = [140 46 71 23];
5788            app.LayerwidthLabel.Text = 'Layer width:';
5789
5790            % Create DirectCheckBox
5791            app.DirectCheckBox = uicheckbox(app.Panel_23);
5792            app.DirectCheckBox.ValueChangedFcn = createCallbackFcn(app,↙
@DirectCheckBoxValueChanged, true);
5793            app.DirectCheckBox.Tooltip = {'L-D type sponge'};
5794            app.DirectCheckBox.Text = ' Direct';
5795            app.DirectCheckBox.Position = [29 107 57 22];
5796            app.DirectCheckBox.Value = true;
5797
5798            % Create FrictionCheckBox
```

```
5799            app.FrictionCheckBox = uicheckbox(app.Panel_23);
5800            app.FrictionCheckBox.ValueChangedFcn = createCallbackFcn(app, ↙
@FrictionCheckBoxValueChanged, true);
5801            app.FrictionCheckBox.Text = ' Friction';
5802            app.FrictionCheckBox.Position = [138 107 65 22];
5803
5804            % Create DiffusionCheckBox
5805            app.DiffusionCheckBox = uicheckbox(app.Panel_23);
5806            app.DiffusionCheckBox.ValueChangedFcn = createCallbackFcn(app, ↙
@DiffusionCheckBoxValueChanged, true);
5807            app.DiffusionCheckBox.Text = ' Diffusion';
5808            app.DiffusionCheckBox.Position = [255 107 72 22];
5809
5810            % Create NorthEditField
5811            app.NorthEditField = uieditfield(app.Panel_23, 'numeric');
5812            app.NorthEditField.Limits = [0 Inf];
5813            app.NorthEditField.ValueDisplayFormat = '%5.1f';
5814            app.NorthEditField.ValueChangedFcn = createCallbackFcn(app, ↙
@NorthEditFieldValueChanged, true);
5815            app.NorthEditField.FontColor = [0.651 0.651 0.651];
5816            app.NorthEditField.Tooltip = {''};
5817            app.NorthEditField.Position = [93 29 61 16];
5818            app.NorthEditField.Value = 100000;
5819
5820            % Create EastEditField
5821            app.EastEditField = uieditfield(app.Panel_23, 'numeric');
5822            app.EastEditField.Limits = [0 Inf];
5823            app.EastEditField.ValueDisplayFormat = '%5.1f';
5824            app.EastEditField.ValueChangedFcn = createCallbackFcn(app, ↙
@EastEditFieldValueChanged, true);
5825            app.EastEditField.FontColor = [0.651 0.651 0.651];
5826            app.EastEditField.Tooltip = {''};
5827            app.EastEditField.Position = [218 29 62 16];
5828            app.EastEditField.Value = 100000;
5829
5830            % Create SouthEditField
5831            app.SouthEditField = uieditfield(app.Panel_23, 'numeric');
5832            app.SouthEditField.Limits = [0 Inf];
5833            app.SouthEditField.ValueDisplayFormat = '%5.1f';
5834            app.SouthEditField.ValueChangedFcn = createCallbackFcn(app, ↙
@SouthEditFieldValueChanged, true);
5835            app.SouthEditField.FontColor = [0.651 0.651 0.651];
5836            app.SouthEditField.Tooltip = {''};
5837            app.SouthEditField.Position = [93 8 61 16];
5838            app.SouthEditField.Value = 100000;
5839
5840            % Create WestEditField
5841            app.WestEditField = uieditfield(app.Panel_23, 'numeric');
5842            app.WestEditField.Limits = [0 Inf];
5843            app.WestEditField.ValueDisplayFormat = '%5.1f';
5844            app.WestEditField.ValueChangedFcn = createCallbackFcn(app, ↙
@WestEditFieldValueChanged, true);
5845            app.WestEditField.FontColor = [0.651 0.651 0.651];
5846            app.WestEditField.Tooltip = {''};
5847            app.WestEditField.Position = [218 8 62 16];
5848            app.WestEditField.Value = 100000;
5849
5850            % Create NorthLabel
```

```
5851             app.NorthLabel = uilabel(app.Panel_23);
5852             app.NorthLabel.Tooltip = {'Unit: meters'};
5853             app.NorthLabel.Position = [53 26 35 22];
5854             app.NorthLabel.Text = 'North';
5855 
5856             % Create EastLabel
5857             app.EastLabel = uilabel(app.Panel_23);
5858             app.EastLabel.Tooltip = {'Unit: meters'};
5859             app.EastLabel.Position = [185 26 29 22];
5860             app.EastLabel.Text = 'East';
5861 
5862             % Create WestLabel
5863             app.WestLabel = uilabel(app.Panel_23);
5864             app.WestLabel.Tooltip = {'Unit: meters'};
5865             app.WestLabel.Position = [185 5 32 22];
5866             app.WestLabel.Text = 'West';
5867 
5868             % Create SouthLabel
5869             app.SouthLabel = uilabel(app.Panel_23);
5870             app.SouthLabel.Tooltip = {'Unit: meters'};
5871             app.SouthLabel.Position = [53 5 37 22];
5872             app.SouthLabel.Text = 'South';
5873 
5874             % Create DecayTypeDropDownLabel
5875             app.DecayTypeDropDownLabel = uilabel(app.Panel_23);
5876             app.DecayTypeDropDownLabel.Position = [29 87 66 22];
5877             app.DecayTypeDropDownLabel.Text = 'Decay type';
5878 
5879             % Create DecayTypeDropDown
5880             app.DecayTypeDropDown = uidropdown(app.Panel_23);
5881             app.DecayTypeDropDown.Items = {'Very mild', 'Mild', 'Sharp'};
5882             app.DecayTypeDropDown.ValueChangedFcn = createCallbackFcn(app, ↙
@DecayTypeDropDownValueChanged, true);
5883             app.DecayTypeDropDown.Position = [29 73 72 16];
5884             app.DecayTypeDropDown.Value = 'Sharp';
5885 
5886             % Create MaximumCdEditFieldLabel
5887             app.MaximumCdEditFieldLabel = uilabel(app.Panel_23);
5888             app.MaximumCdEditFieldLabel.Visible = 'off';
5889             app.MaximumCdEditFieldLabel.Tooltip = {'Cd = drag coefficient'};
5890             app.MaximumCdEditFieldLabel.Position = [138 87 78 22];
5891             app.MaximumCdEditFieldLabel.Text = 'Maximum Cd';
5892 
5893             % Create MaximumCdEditField
5894             app.MaximumCdEditField = uieditfield(app.Panel_23, 'numeric');
5895             app.MaximumCdEditField.Limits = [0 Inf];
5896             app.MaximumCdEditField.ValueDisplayFormat = '%11.3f';
5897             app.MaximumCdEditField.ValueChangedFcn = createCallbackFcn(app, ↙
@MaximumCdEditFieldValueChanged, true);
5898             app.MaximumCdEditField.FontColor = [0.651 0.651 0.651];
5899             app.MaximumCdEditField.Visible = 'off';
5900             app.MaximumCdEditField.Tooltip = {''};
5901             app.MaximumCdEditField.Position = [138 73 51 16];
5902             app.MaximumCdEditField.Value = 10;
5903 
5904             % Create MaximumCspEditFieldLabel
5905             app.MaximumCspEditFieldLabel = uilabel(app.Panel_23);
5906             app.MaximumCspEditFieldLabel.Visible = 'off';
```

```
5907            app.MaximumCspEditFieldLabel.Tooltip = {'Csd = maximum diffusion↙
coefficient'};
5908            app.MaximumCspEditFieldLabel.Position = [255 86 84 22];
5909            app.MaximumCspEditFieldLabel.Text = 'Maximum Csp';
5910
5911            % Create MaximumCspEditField
5912            app.MaximumCspEditField = uieditfield(app.Panel_23, 'numeric');
5913            app.MaximumCspEditField.Limits = [0 Inf];
5914            app.MaximumCspEditField.ValueDisplayFormat = '%11.3f';
5915            app.MaximumCspEditField.ValueChangedFcn = createCallbackFcn(app,↙
@MaximumCspEditFieldValueChanged, true);
5916            app.MaximumCspEditField.FontColor = [0.651 0.651 0.651];
5917            app.MaximumCspEditField.Visible = 'off';
5918            app.MaximumCspEditField.Tooltip = {''};
5919            app.MaximumCspEditField.Position = [255 73 42 16];
5920            app.MaximumCspEditField.Value = 1;
5921
5922            % Create SpongeBoundaryLabel
5923            app.SpongeBoundaryLabel = uilabel(app.InputTab);
5924            app.SpongeBoundaryLabel.BackgroundColor = [0.9412 0.9412 0.9412];
5925            app.SpongeBoundaryLabel.FontSize = 13;
5926            app.SpongeBoundaryLabel.FontWeight = 'bold';
5927            app.SpongeBoundaryLabel.FontColor = [0.0314 0.3686 0.6];
5928            app.SpongeBoundaryLabel.Tooltip = {'Use sponge layer for wave↙
damping'};
5929            app.SpongeBoundaryLabel.Position = [22 192 130 21];
5930            app.SpongeBoundaryLabel.Text = '  Sponge Boundary  ';
5931
5932            % Create Panel_24
5933            app.Panel_24 = uipanel(app.InputTab);
5934            app.Panel_24.AutoResizeChildren = 'off';
5935            app.Panel_24.Position = [10 10 351 39];
5936
5937            % Create ObstacleLoadFile
5938            app.ObstacleLoadFile = uibutton(app.Panel_24, 'push');
5939            app.ObstacleLoadFile.ButtonPushedFcn = createCallbackFcn(app,↙
@Button_19Pushed, true);
5940            app.ObstacleLoadFile.VerticalAlignment = 'top';
5941            app.ObstacleLoadFile.Visible = 'off';
5942            app.ObstacleLoadFile.Tooltip = {''};
5943            app.ObstacleLoadFile.Position = [287 7 19 19];
5944            app.ObstacleLoadFile.Text = '...';
5945
5946            % Create ObstacleFilename
5947            app.ObstacleFilename = uieditfield(app.Panel_24, 'text');
5948            app.ObstacleFilename.Editable = 'off';
5949            app.ObstacleFilename.FontSize = 11;
5950            app.ObstacleFilename.Visible = 'off';
5951            app.ObstacleFilename.Position = [161 8 123 16];
5952
5953            % Create AddobstaclesCheckBox
5954            app.AddobstaclesCheckBox = uicheckbox(app.Panel_24);
5955            app.AddobstaclesCheckBox.ValueChangedFcn = createCallbackFcn(app,↙
@AddobstaclesCheckBoxValueChanged, true);
5956            app.AddobstaclesCheckBox.Tooltip = {'Integrate obstacles with a↙
file format matching the depth data: 1 represents water, and 0 represents dry↙
points.'; ''; 'File size must be the same as the input bathymetry.'};
5957            app.AddobstaclesCheckBox.Text = ' Add obstacles';
```

```
5958            app.AddobstaclesCheckBox.Position = [52 5 102 22];
5959
5960            % Create ObstaclesLabel
5961            app.ObstaclesLabel = uilabel(app.InputTab);
5962            app.ObstaclesLabel.BackgroundColor = [0.9412 0.9412 0.9412];
5963            app.ObstaclesLabel.FontSize = 13;
5964            app.ObstaclesLabel.FontWeight = 'bold';
5965            app.ObstaclesLabel.FontColor = [0.0314 0.3686 0.6];
5966            app.ObstaclesLabel.Position = [22 37 78 21];
5967            app.ObstaclesLabel.Text = '  Obstacles  ';
5968
5969            % Create PhysicsLabel
5970            app.PhysicsLabel = uilabel(app.InputTab);
5971            app.PhysicsLabel.BackgroundColor = [0.9412 0.9412 0.9412];
5972            app.PhysicsLabel.HorizontalAlignment = 'center';
5973            app.PhysicsLabel.FontSize = 13;
5974            app.PhysicsLabel.FontWeight = 'bold';
5975            app.PhysicsLabel.FontColor = [0.0314 0.3686 0.6];
5976            app.PhysicsLabel.Position = [383 627 73 21];
5977            app.PhysicsLabel.Text = '  Physics   ';
5978
5979            % Create Panel_26
5980            app.Panel_26 = uipanel(app.InputTab);
5981            app.Panel_26.AutoResizeChildren = 'off';
5982            app.Panel_26.Position = [371 450 356 60];
5983
5984            % Create ButtonGroup_8
5985            app.ButtonGroup_8 = uibuttongroup(app.Panel_26);
5986            app.ButtonGroup_8.AutoResizeChildren = 'off';
5987            app.ButtonGroup_8.SelectionChangedFcn = createCallbackFcn(app, ↙
@ButtonGroup_8SelectionChanged, true);
5988            app.ButtonGroup_8.BorderType = 'none';
5989            app.ButtonGroup_8.Position = [18 9 159 41];
5990
5991            % Create ConstantCoefficientButton
5992            app.ConstantCoefficientButton = uiradiobutton(app.ButtonGroup_8);
5993            app.ConstantCoefficientButton.Tooltip = {'Set a constant drag↙
friction value throughout the domain'};
5994            app.ConstantCoefficientButton.Text = ' Constant Coefficient';
5995            app.ConstantCoefficientButton.Position = [16 20 135 22];
5996            app.ConstantCoefficientButton.Value = true;
5997
5998            % Create NonconstantButton
5999            app.NonconstantButton = uiradiobutton(app.ButtonGroup_8);
6000            app.NonconstantButton.Tooltip = {'Assign a non-uniform drag↙
friction value across the domain.'};
6001            app.NonconstantButton.Text = ' Nonconstant';
6002            app.NonconstantButton.Position = [16 0 94 22];
6003
6004            % Create FrictionFileEditFieldLabel
6005            app.FrictionFileEditFieldLabel = uilabel(app.Panel_26);
6006            app.FrictionFileEditFieldLabel.HorizontalAlignment = 'right';
6007            app.FrictionFileEditFieldLabel.Visible = 'off';
6008            app.FrictionFileEditFieldLabel.Tooltip = {'Import the friction↙
file.'; ''; 'Dimension should be the same with the input bathymetry.'};
6009            app.FrictionFileEditFieldLabel.Position = [189 9 25 22];
6010            app.FrictionFileEditFieldLabel.Text = 'File';
6011
```

```
6012             % Create FrictionFileEditField
6013             app.FrictionFileEditField = uieditfield(app.Panel_26, 'text');
6014             app.FrictionFileEditField.ValueChangingFcn = createCallbackFcn↙
(app, @FrictionFileEditFieldValueChanging, true);
6015             app.FrictionFileEditField.Editable = 'off';
6016             app.FrictionFileEditField.FontSize = 11;
6017             app.FrictionFileEditField.Visible = 'off';
6018             app.FrictionFileEditField.Tooltip = {'Dimension should be the same↙
with the input bathymetry'};
6019             app.FrictionFileEditField.Position = [222 12 84 16];
6020
6021             % Create Button_10
6022             app.Button_10 = uibutton(app.Panel_26, 'push');
6023             app.Button_10.ButtonPushedFcn = createCallbackFcn(app, ↙
@Button_10Pushed, true);
6024             app.Button_10.VerticalAlignment = 'top';
6025             app.Button_10.Visible = 'off';
6026             app.Button_10.Tooltip = {''};
6027             app.Button_10.Position = [311 11 19 19];
6028             app.Button_10.Text = '...';
6029
6030             % Create ValueEditFieldLabel
6031             app.ValueEditFieldLabel = uilabel(app.Panel_26);
6032             app.ValueEditFieldLabel.HorizontalAlignment = 'right';
6033             app.ValueEditFieldLabel.Tooltip = {'Set to 0 for no friction'};
6034             app.ValueEditFieldLabel.Position = [180 29 34 22];
6035             app.ValueEditFieldLabel.Text = 'Value';
6036
6037             % Create ConstantFrictionValue
6038             app.ConstantFrictionValue = uieditfield(app.Panel_26, 'numeric');
6039             app.ConstantFrictionValue.ValueDisplayFormat = '%8.3f';
6040             app.ConstantFrictionValue.ValueChangedFcn = createCallbackFcn(app, ↙
@ConstantFrictionValueValueChanged, true);
6041             app.ConstantFrictionValue.FontColor = [0.651 0.651 0.651];
6042             app.ConstantFrictionValue.Tooltip = {''};
6043             app.ConstantFrictionValue.Position = [222 32 41 16];
6044
6045             % Create BottomFrictionLabel
6046             app.BottomFrictionLabel = uilabel(app.InputTab);
6047             app.BottomFrictionLabel.BackgroundColor = [0.9412 0.9412 0.9412];
6048             app.BottomFrictionLabel.FontSize = 13;
6049             app.BottomFrictionLabel.FontWeight = 'bold';
6050             app.BottomFrictionLabel.FontColor = [0.0314 0.3686 0.6];
6051             app.BottomFrictionLabel.Tooltip = {''};
6052             app.BottomFrictionLabel.Position = [383 500 114 21];
6053             app.BottomFrictionLabel.Text = '  Bottom Friction  ';
6054
6055             % Create Panel_28
6056             app.Panel_28 = uipanel(app.InputTab);
6057             app.Panel_28.AutoResizeChildren = 'off';
6058             app.Panel_28.Position = [371 108 356 148];
6059
6060             % Create CreatenestedgridboundaryCheckBox
6061             app.CreatenestedgridboundaryCheckBox = uicheckbox(app.Panel_28);
6062             app.CreatenestedgridboundaryCheckBox.ValueChangedFcn = ↙
createCallbackFcn(app, @CreatenestedgridboundaryCheckBoxValueChanged, true);
6063             app.CreatenestedgridboundaryCheckBox.Tooltip = {'Use gauges as↙
points to define the boundary of the subgrid layer'};
```

```
6064             app.CreatenestedgridboundaryCheckBox.Text = 'Create nested grid boundary';
6065             app.CreatenestedgridboundaryCheckBox.WordWrap = 'on';
6066             app.CreatenestedgridboundaryCheckBox.Position = [33 84 182 23];
6067
6068             % Create Button_7
6069             app.Button_7 = uibutton(app.Panel_28, 'push');
6070             app.Button_7.ButtonPushedFcn = createCallbackFcn(app, @Button_7Pushed2, true);
6071             app.Button_7.Enable = 'off';
6072             app.Button_7.Tooltip = {'Import the file containing a list of coordinates representing the gauge locations.'; ''; 'Text file format: Tab-delimited, with columns for latitude and longitude.'};
6073             app.Button_7.Position = [230 114 19 19];
6074             app.Button_7.Text = '...';
6075
6076             % Create LoadFileEditField
6077             app.LoadFileEditField = uieditfield(app.Panel_28, 'text');
6078             app.LoadFileEditField.ValueChangedFcn = createCallbackFcn(app, @LoadFileEditFieldValueChanged, true);
6079             app.LoadFileEditField.ValueChangingFcn = createCallbackFcn(app, @LoadFileEditFieldValueChanging, true);
6080             app.LoadFileEditField.Editable = 'off';
6081             app.LoadFileEditField.FontSize = 11;
6082             app.LoadFileEditField.Enable = 'off';
6083             app.LoadFileEditField.Placeholder = '.shp or .txt';
6084             app.LoadFileEditField.Position = [152 115 74 16];
6085
6086             % Create ImportgaugelistCheckBox
6087             app.ImportgaugelistCheckBox = uicheckbox(app.Panel_28);
6088             app.ImportgaugelistCheckBox.ValueChangedFcn = createCallbackFcn(app, @ImportgaugelistCheckBoxValueChanged, true);
6089             app.ImportgaugelistCheckBox.Tooltip = {'Add stations at specific locations to record wave patterns throughout the simulation'};
6090             app.ImportgaugelistCheckBox.Text = 'Import gauge list';
6091             app.ImportgaugelistCheckBox.Position = [33 112 112 22];
6092
6093             % Create NorthEditField_2
6094             app.NorthEditField_2 = uieditfield(app.Panel_28, 'numeric');
6095             app.NorthEditField_2.ValueDisplayFormat = '%5.4f';
6096             app.NorthEditField_2.ValueChangedFcn = createCallbackFcn(app, @NorthEditField_2ValueChanged, true);
6097             app.NorthEditField_2.FontColor = [0.651 0.651 0.651];
6098             app.NorthEditField_2.Enable = 'off';
6099             app.NorthEditField_2.Tooltip = {'North limit'};
6100             app.NorthEditField_2.Position = [147 37 64 16];
6101
6102             % Create EastEditField_2
6103             app.EastEditField_2 = uieditfield(app.Panel_28, 'numeric');
6104             app.EastEditField_2.ValueDisplayFormat = '%5.4f';
6105             app.EastEditField_2.ValueChangedFcn = createCallbackFcn(app, @EastEditField_2ValueChanged, true);
6106             app.EastEditField_2.FontColor = [0.651 0.651 0.651];
6107             app.EastEditField_2.Enable = 'off';
6108             app.EastEditField_2.Tooltip = {'East limit'};
6109             app.EastEditField_2.Position = [217 24 63 16];
6110
6111             % Create SouthEditField 2
```

```
6112             app.SouthEditField_2 = uieditfield(app.Panel_28, 'numeric');
6113             app.SouthEditField_2.ValueDisplayFormat = '%5.4f';
6114             app.SouthEditField_2.ValueChangedFcn = createCallbackFcn(app, ↙
@SouthEditField_2ValueChanged, true);
6115             app.SouthEditField_2.FontColor = [0.651 0.651 0.651];
6116             app.SouthEditField_2.Enable = 'off';
6117             app.SouthEditField_2.Tooltip = {'South limit'};
6118             app.SouthEditField_2.Position = [147 13 64 16];
6119 
6120             % Create WestEditField_2
6121             app.WestEditField_2 = uieditfield(app.Panel_28, 'numeric');
6122             app.WestEditField_2.ValueDisplayFormat = '%5.4f';
6123             app.WestEditField_2.ValueChangedFcn = createCallbackFcn(app, ↙
@WestEditField_2ValueChanged, true);
6124             app.WestEditField_2.FontColor = [0.651 0.651 0.651];
6125             app.WestEditField_2.Enable = 'off';
6126             app.WestEditField_2.Tooltip = {'West limit'};
6127             app.WestEditField_2.Position = [79 24 63 16];
6128 
6129             % Create BoundarylimitsButton
6130             app.BoundarylimitsButton = uibutton(app.Panel_28, 'push');
6131             app.BoundarylimitsButton.ButtonPushedFcn = createCallbackFcn(app, ↙
@BoundarylimitsButtonPushed, true);
6132             app.BoundarylimitsButton.Enable = 'off';
6133             app.BoundarylimitsButton.Tooltip = {'Extract coordinate and grid↙
spacing information from a tif file. '};
6134             app.BoundarylimitsButton.Position = [79 58 201 22];
6135             app.BoundarylimitsButton.Text = 'Boundary limits:';
6136 
6137             % Create SpacingEditFieldLabel
6138             app.SpacingEditFieldLabel = uilabel(app.Panel_28);
6139             app.SpacingEditFieldLabel.WordWrap = 'on';
6140             app.SpacingEditFieldLabel.Enable = 'off';
6141             app.SpacingEditFieldLabel.Tooltip = {'Set the gauge spacing'};
6142             app.SpacingEditFieldLabel.Position = [216 84 77 22];
6143             app.SpacingEditFieldLabel.Text = 'Grid spacing:';
6144 
6145             % Create SpacingEditField
6146             app.SpacingEditField = uieditfield(app.Panel_28, 'numeric');
6147             app.SpacingEditField.ValueDisplayFormat = '%5.4f';
6148             app.SpacingEditField.ValueChangedFcn = createCallbackFcn(app, ↙
@SpacingEditFieldValueChanged, true);
6149             app.SpacingEditField.FontColor = [0.651 0.651 0.651];
6150             app.SpacingEditField.Enable = 'off';
6151             app.SpacingEditField.Tooltip = {''};
6152             app.SpacingEditField.Position = [294 87 48 16];
6153 
6154             % Create Friction_____________________________Label_5
6155             app.Friction_____________________________Label_5 = uilabel(app.↙
InputTab);
6156             app.Friction_____________________________Label_5.↙
BackgroundColor = [0.9412 0.9412 0.9412];
6157             app.Friction_____________________________Label_5.FontSize = 13;
6158             app.Friction_____________________________Label_5.FontWeight =↙
'bold';
6159             app.Friction_____________________________Label_5.FontColor =↙
[0.0314 0.3686 0.6];
6160             app.Friction                             Label_5.Position =↙
```

```
[383 246 65 21];
6161            app.Friction_____________________________Label_5.Text = '↙
Gauges  ';
6162
6163            % Create Panel_29
6164            app.Panel_29 = uipanel(app.InputTab);
6165            app.Panel_29.AutoResizeChildren = 'off';
6166            app.Panel_29.Position = [371 60 356 37];
6167
6168            % Create Button_14
6169            app.Button_14 = uibutton(app.Panel_29, 'push');
6170            app.Button_14.ButtonPushedFcn = createCallbackFcn(app,↙
@Button_14Pushed, true);
6171            app.Button_14.VerticalAlignment = 'top';
6172            app.Button_14.FontSize = 10;
6173            app.Button_14.FontAngle = 'italic';
6174            app.Button_14.Tooltip = {''};
6175            app.Button_14.Position = [288 7 19 18];
6176            app.Button_14.Text = '...';
6177
6178            % Create SavefilestoLabel
6179            app.SavefilestoLabel = uilabel(app.Panel_29);
6180            app.SavefilestoLabel.Tooltip = {'Specify the directory where↙
''INPUT_FILES'', containing all generated files, will be saved.'};
6181            app.SavefilestoLabel.Position = [37 5 74 22];
6182            app.SavefilestoLabel.Text = 'Save files to:';
6183
6184            % Create SavefilestoEditField
6185            app.SavefilestoEditField = uieditfield(app.Panel_29, 'text');
6186            app.SavefilestoEditField.ValueChangedFcn = createCallbackFcn(app,↙
@SavefilestoEditFieldValueChanged, true);
6187            app.SavefilestoEditField.ValueChangingFcn = createCallbackFcn(app,↙
@SavefilestoEditFieldValueChanging, true);
6188            app.SavefilestoEditField.Editable = 'off';
6189            app.SavefilestoEditField.FontSize = 11;
6190            app.SavefilestoEditField.FontColor = [0 0 1];
6191            app.SavefilestoEditField.Tooltip = {'Specify the directory where↙
''OUTPUT_FILES'' that contians all the files generated in this app will be saved'};
6192            app.SavefilestoEditField.Placeholder = 'Default: Desktop';
6193            app.SavefilestoEditField.Position = [113 8 170 16];
6194
6195            % Create Friction_____________________________Label_6
6196            app.Friction_____________________________Label_6 = uilabel(app.↙
InputTab);
6197            app.Friction_____________________________Label_6.↙
BackgroundColor = [0.9412 0.9412 0.9412];
6198            app.Friction_____________________________Label_6.FontSize = 13;
6199            app.Friction_____________________________Label_6.FontWeight =↙
'bold';
6200            app.Friction_____________________________Label_6.FontColor =↙
[0.0314 0.3686 0.6];
6201            app.Friction_____________________________Label_6.Position =↙
[383 87 73 21];
6202            app.Friction_____________________________Label_6.Text = '↙
Directory  ';
6203
6204            % Create DepthLabel
6205            app.DepthLabel = uilabel(app.InputTab);
```

```
6206            app.DepthLabel.BackgroundColor = [0.9412 0.9412 0.9412];
6207            app.DepthLabel.FontSize = 13;
6208            app.DepthLabel.FontWeight = 'bold';
6209            app.DepthLabel.FontColor = [0.0314 0.3686 0.6];
6210            app.DepthLabel.Position = [21 574 52 21];
6211            app.DepthLabel.Text = '  Depth  ';
6212
6213            % Create DimensionLabel
6214            app.DimensionLabel = uilabel(app.InputTab);
6215            app.DimensionLabel.BackgroundColor = [0.9412 0.9412 0.9412];
6216            app.DimensionLabel.FontSize = 13;
6217            app.DimensionLabel.FontWeight = 'bold';
6218            app.DimensionLabel.FontColor = [0.0314 0.3686 0.6];
6219            app.DimensionLabel.Position = [21 479 86 21];
6220            app.DimensionLabel.Text = '  Dimension  ';
6221
6222            % Create Panel_30
6223            app.Panel_30 = uipanel(app.InputTab);
6224            app.Panel_30.AutoResizeChildren = 'off';
6225            app.Panel_30.Position = [371 329 356 55];
6226
6227            % Create ButtonGroup_23
6228            app.ButtonGroup_23 = uibuttongroup(app.Panel_30);
6229            app.ButtonGroup_23.AutoResizeChildren = 'off';
6230            app.ButtonGroup_23.SelectionChangedFcn = createCallbackFcn(app, ↙
@ButtonGroup_23SelectionChanged, true);
6231            app.ButtonGroup_23.BorderType = 'none';
6232            app.ButtonGroup_23.Position = [26 2 155 42];
6233
6234            % Create ShockwavecapturingButton
6235            app.ShockwavecapturingButton = uiradiobutton(app.ButtonGroup_23);
6236            app.ShockwavecapturingButton.Enable = 'off';
6237            app.ShockwavecapturingButton.Text = ' Shock wave capturing';
6238            app.ShockwavecapturingButton.Position = [7 22 145 22];
6239            app.ShockwavecapturingButton.Value = true;
6240
6241            % Create ViscositybreakingButton
6242            app.ViscositybreakingButton = uiradiobutton(app.ButtonGroup_23);
6243            app.ViscositybreakingButton.Enable = 'off';
6244            app.ViscositybreakingButton.Text = ' Viscosity breaking';
6245            app.ViscositybreakingButton.Position = [7 4 122 22];
6246
6247            % Create C1EditField
6248            app.C1EditField = uieditfield(app.Panel_30, 'numeric');
6249            app.C1EditField.Limits = [0 Inf];
6250            app.C1EditField.ValueDisplayFormat = '%.2f';
6251            app.C1EditField.ValueChangedFcn = createCallbackFcn(app, ↙
@C1EditFieldValueChanged, true);
6252            app.C1EditField.FontColor = [0.651 0.651 0.651];
6253            app.C1EditField.Visible = 'off';
6254            app.C1EditField.Tooltip = {''};
6255            app.C1EditField.Position = [222 27 41 16];
6256            app.C1EditField.Value = 0.65;
6257
6258            % Create C1EditFieldLabel
6259            app.C1EditFieldLabel = uilabel(app.Panel_30);
6260            app.C1EditFieldLabel.HorizontalAlignment = 'right';
6261            app.C1EditFieldLabel.Visible = 'off';
```

```
6262             app.C1EditFieldLabel.Tooltip = {'Parameter value from Kennedy et
al. (2000)'};
6263             app.C1EditFieldLabel.Position = [192 23 25 22];
6264             app.C1EditFieldLabel.Text = 'C1';
6265
6266             % Create C2EditFieldLabel
6267             app.C2EditFieldLabel = uilabel(app.Panel_30);
6268             app.C2EditFieldLabel.HorizontalAlignment = 'right';
6269             app.C2EditFieldLabel.Visible = 'off';
6270             app.C2EditFieldLabel.Tooltip = {'Parameter value from Kennedy et
al. (2000)'};
6271             app.C2EditFieldLabel.Position = [192 4 25 22];
6272             app.C2EditFieldLabel.Text = 'C2';
6273
6274             % Create C2EditField
6275             app.C2EditField = uieditfield(app.Panel_30, 'numeric');
6276             app.C2EditField.Limits = [0 Inf];
6277             app.C2EditField.ValueDisplayFormat = '%.2f';
6278             app.C2EditField.ValueChangedFcn = createCallbackFcn(app,
@C2EditFieldValueChanged, true);
6279             app.C2EditField.FontColor = [0.651 0.651 0.651];
6280             app.C2EditField.Visible = 'off';
6281             app.C2EditField.Tooltip = {''};
6282             app.C2EditField.Position = [222 8 41 16];
6283             app.C2EditField.Value = 0.35;
6284
6285             % Create Friction_____________________________Label_7
6286             app.Friction_____________________________Label_7 = uilabel(app.
InputTab);
6287             app.Friction_____________________________Label_7.
BackgroundColor = [0.9412 0.9412 0.9412];
6288             app.Friction_____________________________Label_7.FontSize = 13;
6289             app.Friction_____________________________Label_7.FontWeight =
'bold';
6290             app.Friction_____________________________Label_7.FontColor =
[0.0314 0.3686 0.6];
6291             app.Friction_____________________________Label_7.Position =
[383 373 109 22];
6292             app.Friction_____________________________Label_7.Text = '  Wave
Breaking ';
6293
6294             % Create GenerateinputtxtButton
6295             app.GenerateinputtxtButton = uibutton(app.InputTab, 'push');
6296             app.GenerateinputtxtButton.ButtonPushedFcn = createCallbackFcn
(app, @GenerateinputtxtButtonPushed, true);
6297             app.GenerateinputtxtButton.BackgroundColor = [0.9216 0.9216
0.9216];
6298             app.GenerateinputtxtButton.FontSize = 15;
6299             app.GenerateinputtxtButton.FontWeight = 'bold';
6300             app.GenerateinputtxtButton.Position = [370 10 357 39];
6301             app.GenerateinputtxtButton.Text = 'Generate input.txt';
6302
6303             % Create Panel_33
6304             app.Panel_33 = uipanel(app.InputTab);
6305             app.Panel_33.AutoResizeChildren = 'off';
6306             app.Panel_33.Position = [371 268 356 50];
6307
6308             % Create UseSmagorinskyCheckBox
```

```
6309             app.UseSmagorinskyCheckBox = uicheckbox(app.Panel_33);
6310             app.UseSmagorinskyCheckBox.ValueChangedFcn = createCallbackFcn↙
(app, @UseSmagorinskyCheckBoxValueChanged2, true);
6311             app.UseSmagorinskyCheckBox.Tooltip = {'Apply a Smagorinsky-like↙
method for subgrid turbulent mixing'};
6312             app.UseSmagorinskyCheckBox.Text = ' Smagorinsky mixing';
6313             app.UseSmagorinskyCheckBox.Position = [33 12 134 22];
6314
6315             % Create TimeintervalsecEditFieldLabel
6316             app.TimeintervalsecEditFieldLabel = uilabel(app.Panel_33);
6317             app.TimeintervalsecEditFieldLabel.Visible = 'off';
6318             app.TimeintervalsecEditFieldLabel.Tooltip = {'Set time interval↙
for calculating values'; ''; 'Unit: seconds'};
6319             app.TimeintervalsecEditFieldLabel.Position = [196 23 74 22];
6320             app.TimeintervalsecEditFieldLabel.Text = 'Time interval';
6321
6322             % Create TimeintervalsecEditField
6323             app.TimeintervalsecEditField = uieditfield(app.Panel_33,↙
'numeric');
6324             app.TimeintervalsecEditField.Limits = [0 Inf];
6325             app.TimeintervalsecEditField.FontColor = [0.651 0.651 0.651];
6326             app.TimeintervalsecEditField.Visible = 'off';
6327             app.TimeintervalsecEditField.Position = [273 26 41 16];
6328             app.TimeintervalsecEditField.Value = 20;
6329
6330             % Create SteadyTimeEditFieldLabel
6331             app.SteadyTimeEditFieldLabel = uilabel(app.Panel_33);
6332             app.SteadyTimeEditFieldLabel.Visible = 'off';
6333             app.SteadyTimeEditFieldLabel.Tooltip = {'Set the initial time for↙
calculating mean values'; ''; 'Unit: seconds'};
6334             app.SteadyTimeEditFieldLabel.Position = [196 3 70 22];
6335             app.SteadyTimeEditFieldLabel.Text = 'Steady time';
6336
6337             % Create SteadyTimeEditField
6338             app.SteadyTimeEditField = uieditfield(app.Panel_33, 'numeric');
6339             app.SteadyTimeEditField.Limits = [0 Inf];
6340             app.SteadyTimeEditField.FontColor = [0.651 0.651 0.651];
6341             app.SteadyTimeEditField.Visible = 'off';
6342             app.SteadyTimeEditField.Position = [273 6 41 16];
6343
6344             % Create Friction_____________________________Label_10
6345             app.Friction_____________________________Label_10 = uilabel↙
(app.InputTab);
6346             app.Friction_____________________________Label_10.↙
BackgroundColor = [0.9412 0.9412 0.9412];
6347             app.Friction_____________________________Label_10.FontSize =↙
13;
6348             app.Friction_____________________________Label_10.FontWeight =↙
'bold';
6349             app.Friction_____________________________Label_10.FontColor =↙
[0.0314 0.3686 0.6];
6350             app.Friction_____________________________Label_10.Position =↙
[383 306 116 22];
6351             app.Friction_____________________________Label_10.Text = '↙
Turbulent Mixing';
6352
6353             % Create InitialConditionTab
6354             app.InitialConditionTab = uitab(app.TabGroup);
```

```
6355             app.InitialConditionTab.AutoResizeChildren = 'off';
6356             app.InitialConditionTab.Title = 'Initial Condition';
6357             app.InitialConditionTab.BackgroundColor = [0.9412 0.9412 0.9412];
6358             app.InitialConditionTab.Interruptible = 'off';
6359 
6360             % Create TabGroup2
6361             app.TabGroup2 = uitabgroup(app.InitialConditionTab);
6362             app.TabGroup2.AutoResizeChildren = 'off';
6363             app.TabGroup2.TabLocation = 'left';
6364             app.TabGroup2.SelectionChangedFcn = createCallbackFcn(app,↙
@TabGroup2SelectionChanged, true);
6365             app.TabGroup2.Position = [-69 84 806 515];
6366 
6367             % Create Tab_5
6368             app.Tab_5 = uitab(app.TabGroup2);
6369             app.Tab_5.AutoResizeChildren = 'off';
6370 
6371             % Create Tab
6372             app.Tab = uitab(app.TabGroup2);
6373             app.Tab.AutoResizeChildren = 'off';
6374 
6375             % Create Label_9
6376             app.Label_9 = uilabel(app.Tab);
6377             app.Label_9.BackgroundColor = [0.651 0.651 0.651];
6378             app.Label_9.Position = [1 495 738 22];
6379             app.Label_9.Text = '';
6380 
6381             % Create ImportUVZfilesLabel
6382             app.ImportUVZfilesLabel = uilabel(app.Tab);
6383             app.ImportUVZfilesLabel.FontSize = 14;
6384             app.ImportUVZfilesLabel.FontWeight = 'bold';
6385             app.ImportUVZfilesLabel.FontColor = [1 1 1];
6386             app.ImportUVZfilesLabel.Position = [24 495 131 22];
6387             app.ImportUVZfilesLabel.Text = 'Import  U, V, Z files';
6388 
6389             % Create Button_32
6390             app.Button_32 = uibutton(app.Tab, 'push');
6391             app.Button_32.ButtonPushedFcn = createCallbackFcn(app,↙
@Button_4Pushed, true);
6392             app.Button_32.VerticalAlignment = 'top';
6393             app.Button_32.Tooltip = {'Import the Z file. '; ''; 'Accepted↙
format: .txt'; ''; 'If you only have .tif file, convert this file to .txt in↙
''Input'' tab > Depth > Import Data'; ''; 'The values in the .txt file should↙
include decimal places.'};
6394             app.Button_32.Position = [523 371 22 22];
6395             app.Button_32.Text = '...';
6396 
6397             % Create Button_34
6398             app.Button_34 = uibutton(app.Tab, 'push');
6399             app.Button_34.ButtonPushedFcn = createCallbackFcn(app,↙
@Button_5Pushed, true);
6400             app.Button_34.Tooltip = {'Import the U vector file. '; '';↙
'Accepted format: .txt'; ''; 'If you only have .tif file, convert this file to .txt↙
in ''Input'' tab > Depth > Import Data'; ''; 'The values in the .txt file should↙
include decimal places.'};
6401             app.Button_34.Position = [523 339 22 22];
6402             app.Button_34.Text = '...';
6403 
```

```
6404             % Create SurfaceHeightEditFieldLabel
6405             app.SurfaceHeightEditFieldLabel = uilabel(app.Tab);
6406             app.SurfaceHeightEditFieldLabel.HorizontalAlignment = 'right';
6407             app.SurfaceHeightEditFieldLabel.Tooltip = {'Import file containing↙
the sea surface displacement matrix.'; ''; 'The values in the .txt file should↙
include decimal places.'};
6408             app.SurfaceHeightEditFieldLabel.Position = [196 371 85 22];
6409             app.SurfaceHeightEditFieldLabel.Text = 'Surface Height';
6410
6411             % Create SurfaceHeightEditField
6412             app.SurfaceHeightEditField = uieditfield(app.Tab, 'text');
6413             app.SurfaceHeightEditField.ValueChangingFcn = createCallbackFcn↙
(app, @SurfaceHeightEditFieldValueChanging, true);
6414             app.SurfaceHeightEditField.Editable = 'off';
6415             app.SurfaceHeightEditField.FontSize = 11;
6416             app.SurfaceHeightEditField.Tooltip = {''};
6417             app.SurfaceHeightEditField.Position = [307 371 207 22];
6418
6419             % Create UVelocityEditFieldLabel
6420             app.UVelocityEditFieldLabel = uilabel(app.Tab);
6421             app.UVelocityEditFieldLabel.HorizontalAlignment = 'right';
6422             app.UVelocityEditFieldLabel.Tooltip = {'Import file with x-↙
direction velocity matrix.'; ''; 'The values in the .txt file should include↙
decimal places.'};
6423             app.UVelocityEditFieldLabel.Position = [196 339 59 22];
6424             app.UVelocityEditFieldLabel.Text = 'U Velocity';
6425
6426             % Create UVelocityEditField
6427             app.UVelocityEditField = uieditfield(app.Tab, 'text');
6428             app.UVelocityEditField.ValueChangingFcn = createCallbackFcn(app,↙
@UVelocityEditFieldValueChanging, true);
6429             app.UVelocityEditField.Editable = 'off';
6430             app.UVelocityEditField.FontSize = 11;
6431             app.UVelocityEditField.Tooltip = {''};
6432             app.UVelocityEditField.Position = [307 339 207 22];
6433
6434             % Create VVelocityEditFieldLabel
6435             app.VVelocityEditFieldLabel = uilabel(app.Tab);
6436             app.VVelocityEditFieldLabel.HorizontalAlignment = 'right';
6437             app.VVelocityEditFieldLabel.Tooltip = {'Import file with y-↙
direction velocity matrix.'; ''; 'The values in the .txt file should include↙
decimal places.'};
6438             app.VVelocityEditFieldLabel.Position = [197 307 58 22];
6439             app.VVelocityEditFieldLabel.Text = 'V Velocity';
6440
6441             % Create VVelocityEditField
6442             app.VVelocityEditField = uieditfield(app.Tab, 'text');
6443             app.VVelocityEditField.ValueChangingFcn = createCallbackFcn(app,↙
@VVelocityEditFieldValueChanging, true);
6444             app.VVelocityEditField.Editable = 'off';
6445             app.VVelocityEditField.FontSize = 11;
6446             app.VVelocityEditField.Tooltip = {''};
6447             app.VVelocityEditField.Position = [307 306 207 22];
6448
6449             % Create Button_33
6450             app.Button_33 = uibutton(app.Tab, 'push');
6451             app.Button_33.ButtonPushedFcn = createCallbackFcn(app,↙
@Button_6Pushed, true);
```

```
6452             app.Button_33.Tooltip = {'Import the V vector file. '; ''; 'Accepted format: .txt'; ''; 'If you only have .tif file, convert this file to .txt in ''Input'' tab > Depth > Import Data'; ''; 'The values in the .txt file should include decimal places.'};
6453             app.Button_33.Position = [523 306 22 22];
6454             app.Button_33.Text = '...';
6455
6456             % Create Tab_2
6457             app.Tab_2 = uitab(app.TabGroup2);
6458             app.Tab_2.AutoResizeChildren = 'off';
6459
6460             % Create Label_10
6461             app.Label_10 = uilabel(app.Tab_2);
6462             app.Label_10.BackgroundColor = [0.651 0.651 0.651];
6463             app.Label_10.Position = [1 495 751 22];
6464             app.Label_10.Text = '';
6465
6466             % Create WavemakerParametersLabel_2
6467             app.WavemakerParametersLabel_2 = uilabel(app.Tab_2);
6468             app.WavemakerParametersLabel_2.FontSize = 14;
6469             app.WavemakerParametersLabel_2.FontWeight = 'bold';
6470             app.WavemakerParametersLabel_2.FontColor = [1 1 1];
6471             app.WavemakerParametersLabel_2.Position = [24 495 165 22];
6472             app.WavemakerParametersLabel_2.Text = 'Wavemaker Parameters';
6473
6474             % Create Button_20
6475             app.Button_20 = uibutton(app.Tab_2, 'push');
6476             app.Button_20.ButtonPushedFcn = createCallbackFcn(app, @Button_20Pushed, true);
6477             app.Button_20.Enable = 'off';
6478             app.Button_20.Tooltip = {''};
6479             app.Button_20.Position = [404 77 22 18];
6480             app.Button_20.Text = '...';
6481
6482             % Create WavemakertypeDropDownLabel
6483             app.WavemakertypeDropDownLabel = uilabel(app.Tab_2);
6484             app.WavemakertypeDropDownLabel.HorizontalAlignment = 'right';
6485             app.WavemakertypeDropDownLabel.Position = [161 459 95 22];
6486             app.WavemakertypeDropDownLabel.Text = 'Wavemaker type';
6487
6488             % Create WavemakertypeDropDown
6489             app.WavemakertypeDropDown = uidropdown(app.Tab_2);
6490             app.WavemakertypeDropDown.Items = {'See options', 'INI_REC', 'INI_GAUSSIAN', 'LEF_SOL', 'INI_SOL', 'JON_1D', 'JON_2D', 'WK_IRR', 'WK_REG', 'WK_TIME_SERIES', 'WK_DATA2D', 'TMA_1D'};
6491             app.WavemakertypeDropDown.DropDownOpeningFcn = createCallbackFcn(app, @WavemakertypeDropDownOpening, true);
6492             app.WavemakertypeDropDown.ValueChangedFcn = createCallbackFcn(app, @WavemakertypeDropDownValueChanged, true);
6493             app.WavemakertypeDropDown.Position = [310 460 134 22];
6494             app.WavemakertypeDropDown.Value = 'See options';
6495
6496             % Create AmplitudemEditFieldLabel
6497             app.AmplitudemEditFieldLabel = uilabel(app.Tab_2);
6498             app.AmplitudemEditFieldLabel.HorizontalAlignment = 'right';
6499             app.AmplitudemEditFieldLabel.FontColor = [0.149 0.149 0.149];
6500             app.AmplitudemEditFieldLabel.Tooltip = {'Height of the initial sea surface displacement'};
```

```
6501             app.AmplitudemEditFieldLabel.Position = [161 427 80 22];
6502             app.AmplitudemEditFieldLabel.Text = 'Amplitude (m)';
6503 
6504             % Create AmplitudemEditField
6505             app.AmplitudemEditField = uieditfield(app.Tab_2, 'numeric');
6506             app.AmplitudemEditField.Limits = [0 Inf];
6507             app.AmplitudemEditField.ValueDisplayFormat = '%11.2f';
6508             app.AmplitudemEditField.ValueChangedFcn = createCallbackFcn(app,↙
@AmplitudemEditFieldValueChanged, true);
6509             app.AmplitudemEditField.FontColor = [0.8 0.8 0.8];
6510             app.AmplitudemEditField.Enable = 'off';
6511             app.AmplitudemEditField.Tooltip = {''};
6512             app.AmplitudemEditField.Position = [310 429 70 18];
6513 
6514             % Create WaterdepthmEditFieldLabel
6515             app.WaterdepthmEditFieldLabel = uilabel(app.Tab_2);
6516             app.WaterdepthmEditFieldLabel.HorizontalAlignment = 'right';
6517             app.WaterdepthmEditFieldLabel.FontColor = [0.149 0.149 0.149];
6518             app.WaterdepthmEditFieldLabel.Tooltip = {'Water depth at the↙
location of the wavemaker'};
6519             app.WaterdepthmEditFieldLabel.Position = [161 395 91 22];
6520             app.WaterdepthmEditFieldLabel.Text = 'Water depth (m)';
6521 
6522             % Create WaterdepthmEditField
6523             app.WaterdepthmEditField = uieditfield(app.Tab_2, 'numeric');
6524             app.WaterdepthmEditField.Limits = [0 Inf];
6525             app.WaterdepthmEditField.ValueDisplayFormat = '%11.2f';
6526             app.WaterdepthmEditField.ValueChangedFcn = createCallbackFcn(app,↙
@WaterdepthmEditFieldValueChanged, true);
6527             app.WaterdepthmEditField.FontColor = [0.8 0.8 0.8];
6528             app.WaterdepthmEditField.Enable = 'off';
6529             app.WaterdepthmEditField.Tooltip = {''};
6530             app.WaterdepthmEditField.Position = [310 397 70 18];
6531 
6532             % Create LagtimesecEditFieldLabel
6533             app.LagtimesecEditFieldLabel = uilabel(app.Tab_2);
6534             app.LagtimesecEditFieldLabel.HorizontalAlignment = 'right';
6535             app.LagtimesecEditFieldLabel.FontColor = [0.149 0.149 0.149];
6536             app.LagtimesecEditFieldLabel.Tooltip = {'Lag time associated with↙
the solitary wave generated at the left boundary'};
6537             app.LagtimesecEditFieldLabel.Position = [161 139 80 22];
6538             app.LagtimesecEditFieldLabel.Text = 'Lag time (sec)';
6539 
6540             % Create LagtimesecEditField
6541             app.LagtimesecEditField = uieditfield(app.Tab_2, 'numeric');
6542             app.LagtimesecEditField.Limits = [0 Inf];
6543             app.LagtimesecEditField.ValueDisplayFormat = '%.2f';
6544             app.LagtimesecEditField.ValueChangedFcn = createCallbackFcn(app,↙
@LagtimesecEditFieldValueChanged, true);
6545             app.LagtimesecEditField.FontColor = [0.8 0.8 0.8];
6546             app.LagtimesecEditField.Enable = 'off';
6547             app.LagtimesecEditField.Tooltip = {''};
6548             app.LagtimesecEditField.Position = [310 141 70 18];
6549 
6550             % Create XcoordinatemEditFieldLabel
6551             app.XcoordinatemEditFieldLabel = uilabel(app.Tab_2);
6552             app.XcoordinatemEditFieldLabel.HorizontalAlignment = 'right';
6553             app.XcoordinatemEditFieldLabel.FontColor = [0.149 0.149 0.149];
```

```
6554             app.XcoordinatemEditFieldLabel.Tooltip = {'Coordinate in X↙
direction.'; ''; 'For INI_REC, it refers to the location of the center of the↙
rectangular hump'};
6555             app.XcoordinatemEditFieldLabel.Position = [161 363 93 22];
6556             app.XcoordinatemEditFieldLabel.Text = 'X coordinate (m)';
6557
6558             % Create XcoordinatemEditField
6559             app.XcoordinatemEditField = uieditfield(app.Tab_2, 'numeric');
6560             app.XcoordinatemEditField.ValueDisplayFormat = '%.4f';
6561             app.XcoordinatemEditField.ValueChangedFcn = createCallbackFcn(app,↙
@XcoordinatemEditFieldValueChanged, true);
6562             app.XcoordinatemEditField.FontColor = [0.8 0.8 0.8];
6563             app.XcoordinatemEditField.Enable = 'off';
6564             app.XcoordinatemEditField.Tooltip = {''};
6565             app.XcoordinatemEditField.Position = [310 365 70 18];
6566
6567             % Create YcoordinatemEditFieldLabel
6568             app.YcoordinatemEditFieldLabel = uilabel(app.Tab_2);
6569             app.YcoordinatemEditFieldLabel.HorizontalAlignment = 'right';
6570             app.YcoordinatemEditFieldLabel.FontColor = [0.149 0.149 0.149];
6571             app.YcoordinatemEditFieldLabel.Tooltip = {'Coordinate in Y↙
direction'; ''; 'For INI_REC, it refers to the location of the center of the↙
rectangular hump'};
6572             app.YcoordinatemEditFieldLabel.Position = [161 331 94 22];
6573             app.YcoordinatemEditFieldLabel.Text = 'Y coordinate (m)';
6574
6575             % Create YcoordinatemEditField
6576             app.YcoordinatemEditField = uieditfield(app.Tab_2, 'numeric');
6577             app.YcoordinatemEditField.ValueDisplayFormat = '%.4f';
6578             app.YcoordinatemEditField.ValueChangedFcn = createCallbackFcn(app,↙
@YcoordinatemEditFieldValueChanged, true);
6579             app.YcoordinatemEditField.FontColor = [0.8 0.8 0.8];
6580             app.YcoordinatemEditField.Enable = 'off';
6581             app.YcoordinatemEditField.Tooltip = {''};
6582             app.YcoordinatemEditField.Position = [310 333 70 18];
6583
6584             % Create WidthmEditFieldLabel
6585             app.WidthmEditFieldLabel = uilabel(app.Tab_2);
6586             app.WidthmEditFieldLabel.HorizontalAlignment = 'right';
6587             app.WidthmEditFieldLabel.FontColor = [0.149 0.149 0.149];
6588             app.WidthmEditFieldLabel.Tooltip = {'For INI_REC or INI_GAUSSIAN,↙
it refers to the width of the rectangular hump '; ''; 'For WK_REG, it is the width↙
in Y direction'};
6589             app.WidthmEditFieldLabel.Position = [161 299 57 22];
6590             app.WidthmEditFieldLabel.Text = 'Width (m)';
6591
6592             % Create WidthmEditField
6593             app.WidthmEditField = uieditfield(app.Tab_2, 'numeric');
6594             app.WidthmEditField.ValueDisplayFormat = '%.2f';
6595             app.WidthmEditField.ValueChangedFcn = createCallbackFcn(app,↙
@WidthmEditFieldValueChanged, true);
6596             app.WidthmEditField.FontColor = [0.8 0.8 0.8];
6597             app.WidthmEditField.Enable = 'off';
6598             app.WidthmEditField.Tooltip = {''};
6599             app.WidthmEditField.Position = [310 301 70 18];
6600
6601             % Create TimerampsecEditFieldLabel
6602             app.TimerampsecEditFieldLabel = uilabel(app.Tab_2);
```

```
6603            app.TimerampsecEditFieldLabel.HorizontalAlignment = 'right';
6604            app.TimerampsecEditFieldLabel.FontColor = [0.149 0.149 0.149];
6605            app.TimerampsecEditFieldLabel.Tooltip = {'The period during which
wave height gradually increases from zero to its maximum'};
6606            app.TimerampsecEditFieldLabel.Position = [161 203 91 22];
6607            app.TimerampsecEditFieldLabel.Text = 'Time ramp (sec)';
6608
6609            % Create TimerampsecEditField
6610            app.TimerampsecEditField = uieditfield(app.Tab_2, 'numeric');
6611            app.TimerampsecEditField.Limits = [0 Inf];
6612            app.TimerampsecEditField.ValueDisplayFormat = '%.2f';
6613            app.TimerampsecEditField.ValueChangedFcn = createCallbackFcn(app,
@TimerampsecEditFieldValueChanged, true);
6614            app.TimerampsecEditField.FontColor = [0.8 0.8 0.8];
6615            app.TimerampsecEditField.Enable = 'off';
6616            app.TimerampsecEditField.Tooltip = {''};
6617            app.TimerampsecEditField.Position = [310 205 70 18];
6618
6619            % Create WidthDeltaEditFieldLabel
6620            app.WidthDeltaEditFieldLabel = uilabel(app.Tab_2);
6621            app.WidthDeltaEditFieldLabel.HorizontalAlignment = 'right';
6622            app.WidthDeltaEditFieldLabel.FontColor = [0.149 0.149 0.149];
6623            app.WidthDeltaEditFieldLabel.Tooltip = {'Delta width parameter';
'';'Default values: 0.3 ~ 0.6'};
6624            app.WidthDeltaEditFieldLabel.Position = [161 267 68 22];
6625            app.WidthDeltaEditFieldLabel.Text = 'Width Delta';
6626
6627            % Create WidthDeltaEditField
6628            app.WidthDeltaEditField = uieditfield(app.Tab_2, 'numeric');
6629            app.WidthDeltaEditField.ValueDisplayFormat = '%.2f';
6630            app.WidthDeltaEditField.ValueChangedFcn = createCallbackFcn(app,
@WidthDeltaEditFieldValueChanged, true);
6631            app.WidthDeltaEditField.FontColor = [0.8 0.8 0.8];
6632            app.WidthDeltaEditField.Enable = 'off';
6633            app.WidthDeltaEditField.Tooltip = {''};
6634            app.WidthDeltaEditField.Position = [310 269 70 18];
6635
6636            % Create PeriodsecEditFieldLabel
6637            app.PeriodsecEditFieldLabel = uilabel(app.Tab_2);
6638            app.PeriodsecEditFieldLabel.HorizontalAlignment = 'right';
6639            app.PeriodsecEditFieldLabel.FontColor = [0.149 0.149 0.149];
6640            app.PeriodsecEditFieldLabel.Tooltip = {'Time period of the regular
wave'};
6641            app.PeriodsecEditFieldLabel.Position = [161 171 69 22];
6642            app.PeriodsecEditFieldLabel.Text = 'Period (sec)';
6643
6644            % Create PeriodsecEditField
6645            app.PeriodsecEditField = uieditfield(app.Tab_2, 'numeric');
6646            app.PeriodsecEditField.Limits = [0 Inf];
6647            app.PeriodsecEditField.ValueDisplayFormat = '%.2f';
6648            app.PeriodsecEditField.ValueChangedFcn = createCallbackFcn(app,
@PeriodsecEditFieldValueChanged, true);
6649            app.PeriodsecEditField.FontColor = [0.8 0.8 0.8];
6650            app.PeriodsecEditField.Enable = 'off';
6651            app.PeriodsecEditField.Tooltip = {''};
6652            app.PeriodsecEditField.Position = [310 173 70 18];
6653
6654            % Create ThetadegreesEditFieldLabel
```

```
6655            app.ThetadegreesEditFieldLabel = uilabel(app.Tab_2);
6656            app.ThetadegreesEditFieldLabel.HorizontalAlignment = 'right';
6657            app.ThetadegreesEditFieldLabel.FontColor = [0.149 0.149 0.149];
6658            app.ThetadegreesEditFieldLabel.Tooltip = {'Direction of regular wave'};
6659            app.ThetadegreesEditFieldLabel.Position = [161 235 88 22];
6660            app.ThetadegreesEditFieldLabel.Text = 'Theta (degrees)';
6661
6662            % Create ThetadegreesEditField
6663            app.ThetadegreesEditField = uieditfield(app.Tab_2, 'numeric');
6664            app.ThetadegreesEditField.ValueDisplayFormat = '%.2f';
6665            app.ThetadegreesEditField.ValueChangedFcn = createCallbackFcn(app, @ThetadegreesEditFieldValueChanged, true);
6666            app.ThetadegreesEditField.FontColor = [0.8 0.8 0.8];
6667            app.ThetadegreesEditField.Enable = 'off';
6668            app.ThetadegreesEditField.Tooltip = {''};
6669            app.ThetadegreesEditField.Position = [310 237 70 18];
6670
6671            % Create WavecomponentfileEditFieldLabel
6672            app.WavecomponentfileEditFieldLabel = uilabel(app.Tab_2);
6673            app.WavecomponentfileEditFieldLabel.HorizontalAlignment = 'right';
6674            app.WavecomponentfileEditFieldLabel.FontColor = [0.149 0.149 0.149];
6675            app.WavecomponentfileEditFieldLabel.Position = [161 75 118 22];
6676            app.WavecomponentfileEditFieldLabel.Text = 'Wave component file';
6677
6678            % Create WavecomponentfileEditField
6679            app.WavecomponentfileEditField = uieditfield(app.Tab_2, 'text');
6680            app.WavecomponentfileEditField.ValueChangedFcn = createCallbackFcn(app, @WavecomponentfileEditFieldValueChanged, true);
6681            app.WavecomponentfileEditField.Editable = 'off';
6682            app.WavecomponentfileEditField.FontColor = [0.8 0.8 0.8];
6683            app.WavecomponentfileEditField.Enable = 'off';
6684            app.WavecomponentfileEditField.Tooltip = {'Import the file that contains the wave component'};
6685            app.WavecomponentfileEditField.Position = [311 77 89 18];
6686
6687            % Create WavecomponentcountEditFieldLabel
6688            app.WavecomponentcountEditFieldLabel = uilabel(app.Tab_2);
6689            app.WavecomponentcountEditFieldLabel.HorizontalAlignment = 'right';
6690            app.WavecomponentcountEditFieldLabel.FontColor = [0.149 0.149 0.149];
6691            app.WavecomponentcountEditFieldLabel.Tooltip = {'Number of the wave components'};
6692            app.WavecomponentcountEditFieldLabel.Position = [161 107 133 22];
6693            app.WavecomponentcountEditFieldLabel.Text = 'Wave component count';
6694
6695            % Create WaveCompCountEditField
6696            app.WaveCompCountEditField = uieditfield(app.Tab_2, 'numeric');
6697            app.WaveCompCountEditField.ValueDisplayFormat = '%.2f';
6698            app.WaveCompCountEditField.FontColor = [0.8 0.8 0.8];
6699            app.WaveCompCountEditField.Enable = 'off';
6700            app.WaveCompCountEditField.Tooltip = {''};
6701            app.WaveCompCountEditField.Position = [311 109 70 18];
6702
6703            % Create FrequencyLabel
```

```
6704             app.FrequencyLabel = uilabel(app.Tab_2);
6705             app.FrequencyLabel.Position = [166 13 62 22];
6706             app.FrequencyLabel.Text = 'Frequency';
6707 
6708             % Create FrequencyPeakLabel
6709             app.FrequencyPeakLabel = uilabel(app.Tab_2);
6710             app.FrequencyPeakLabel.HorizontalAlignment = 'right';
6711             app.FrequencyPeakLabel.Position = [550 13 32 22];
6712             app.FrequencyPeakLabel.Text = 'Peak';
6713 
6714             % Create PeakEditField
6715             app.PeakEditField = uieditfield(app.Tab_2, 'numeric');
6716             app.PeakEditField.ValueDisplayFormat = '%11.2f';
6717             app.PeakEditField.ValueChangedFcn = createCallbackFcn(app, ↙
@PeakEditFieldValueChanged, true);
6718             app.PeakEditField.FontColor = [0.8 0.8 0.8];
6719             app.PeakEditField.Enable = 'off';
6720             app.PeakEditField.Position = [588 15 48 18];
6721 
6722             % Create MinimumEditFieldLabel
6723             app.MinimumEditFieldLabel = uilabel(app.Tab_2);
6724             app.MinimumEditFieldLabel.HorizontalAlignment = 'right';
6725             app.MinimumEditFieldLabel.Position = [307 13 55 22];
6726             app.MinimumEditFieldLabel.Text = 'Minimum';
6727 
6728             % Create MinimumEditField
6729             app.MinimumEditField = uieditfield(app.Tab_2, 'numeric');
6730             app.MinimumEditField.ValueDisplayFormat = '%11.2f';
6731             app.MinimumEditField.ValueChangedFcn = createCallbackFcn(app, ↙
@MinimumEditFieldValueChanged, true);
6732             app.MinimumEditField.FontColor = [0.8 0.8 0.8];
6733             app.MinimumEditField.Enable = 'off';
6734             app.MinimumEditField.Position = [369 15 48 18];
6735 
6736             % Create MaximumEditFieldLabel
6737             app.MaximumEditFieldLabel = uilabel(app.Tab_2);
6738             app.MaximumEditFieldLabel.HorizontalAlignment = 'right';
6739             app.MaximumEditFieldLabel.Position = [427 13 58 22];
6740             app.MaximumEditFieldLabel.Text = 'Maximum';
6741 
6742             % Create MaximumEditField
6743             app.MaximumEditField = uieditfield(app.Tab_2, 'numeric');
6744             app.MaximumEditField.ValueDisplayFormat = '%11.2f';
6745             app.MaximumEditField.ValueChangedFcn = createCallbackFcn(app, ↙
@MaximumEditFieldValueChanged, true);
6746             app.MaximumEditField.FontColor = [0.8 0.8 0.8];
6747             app.MaximumEditField.Enable = 'off';
6748             app.MaximumEditField.Position = [491 14 48 18];
6749 
6750             % Create WaveHeightmEditFieldLabel
6751             app.WaveHeightmEditFieldLabel = uilabel(app.Tab_2);
6752             app.WaveHeightmEditFieldLabel.HorizontalAlignment = 'right';
6753             app.WaveHeightmEditFieldLabel.Tooltip = {'Significant wave height↙
in Wei and Kirby (1999)'};
6754             app.WaveHeightmEditFieldLabel.Position = [161 44 93 22];
6755             app.WaveHeightmEditFieldLabel.Text = 'Wave Height (m)';
6756 
6757             % Create WaveHeightmEditField
```

```
6758             app.WaveHeightmEditField = uieditfield(app.Tab_2, 'numeric');
6759             app.WaveHeightmEditField.ValueDisplayFormat = '%11.2f';
6760             app.WaveHeightmEditField.ValueChangedFcn = createCallbackFcn(app, ↙
@WaveHeightmEditFieldValueChanged, true);
6761             app.WaveHeightmEditField.FontColor = [0.8 0.8 0.8];
6762             app.WaveHeightmEditField.Enable = 'off';
6763             app.WaveHeightmEditField.Position = [311 46 48 18];
6764 
6765             % Create Tab_4
6766             app.Tab_4 = uitab(app.TabGroup2);
6767             app.Tab_4.AutoResizeChildren = 'off';
6768 
6769             % Create Label_11
6770             app.Label_11 = uilabel(app.Tab_4);
6771             app.Label_11.BackgroundColor = [0.651 0.651 0.651];
6772             app.Label_11.Position = [1 495 738 22];
6773             app.Label_11.Text = '';
6774 
6775             % Create MeteotsunamiLabel
6776             app.MeteotsunamiLabel = uilabel(app.Tab_4);
6777             app.MeteotsunamiLabel.FontSize = 14;
6778             app.MeteotsunamiLabel.FontWeight = 'bold';
6779             app.MeteotsunamiLabel.FontColor = [1 1 1];
6780             app.MeteotsunamiLabel.Position = [24 495 101 22];
6781             app.MeteotsunamiLabel.Text = 'Meteotsunami';
6782 
6783             % Create Button_29
6784             app.Button_29 = uibutton(app.Tab_4, 'push');
6785             app.Button_29.ButtonPushedFcn = createCallbackFcn(app, ↙
@Button_29Pushed, true);
6786             app.Button_29.Enable = 'off';
6787             app.Button_29.Tooltip = {''};
6788             app.Button_29.Position = [575 345 22 20];
6789             app.Button_29.Text = '...';
6790 
6791             % Create Button_30
6792             app.Button_30 = uibutton(app.Tab_4, 'push');
6793             app.Button_30.ButtonPushedFcn = createCallbackFcn(app, ↙
@Button_30Pushed, true);
6794             app.Button_30.Enable = 'off';
6795             app.Button_30.Tooltip = {''};
6796             app.Button_30.Position = [575 226 22 20];
6797             app.Button_30.Text = '...';
6798 
6799             % Create WindforceSwitchLabel
6800             app.WindforceSwitchLabel = uilabel(app.Tab_4);
6801             app.WindforceSwitchLabel.HorizontalAlignment = 'center';
6802             app.WindforceSwitchLabel.Tooltip = {'Use wind effect'};
6803             app.WindforceSwitchLabel.Position = [81 303 64 22];
6804             app.WindforceSwitchLabel.Text = 'Wind force';
6805 
6806             % Create WindforceSwitch
6807             app.WindforceSwitch = uiswitch(app.Tab_4, 'slider');
6808             app.WindforceSwitch.ValueChangedFcn = createCallbackFcn(app, ↙
@WindforceSwitchValueChanged, true);
6809             app.WindforceSwitch.Tooltip = {''};
6810             app.WindforceSwitch.Position = [243 307 32 14];
6811 
```

```
6812             % Create AirpressureSwitchLabel
6813             app.AirpressureSwitchLabel = uilabel(app.Tab_4);
6814             app.AirpressureSwitchLabel.HorizontalAlignment = 'center';
6815             app.AirpressureSwitchLabel.Position = [81 429 70 22];
6816             app.AirpressureSwitchLabel.Text = 'Air pressure';
6817 
6818             % Create AirpressureSwitch
6819             app.AirpressureSwitch = uiswitch(app.Tab_4, 'slider');
6820             app.AirpressureSwitch.ValueChangedFcn = createCallbackFcn(app, ↙
@AirpressureSwitchValueChanged, true);
6821             app.AirpressureSwitch.Tooltip = {'Use air pressure effect'};
6822             app.AirpressureSwitch.Position = [243 433 32 14];
6823             app.AirpressureSwitch.Value = 'On';
6824 
6825             % Create WindstresscoefficientEditField
6826             app.WindstresscoefficientEditField = uieditfield(app.Tab_4, ↙
'numeric');
6827             app.WindstresscoefficientEditField.Limits = [0 Inf];
6828             app.WindstresscoefficientEditField.ValueDisplayFormat = '%11.3g';
6829             app.WindstresscoefficientEditField.ValueChangedFcn = ↙
createCallbackFcn(app, @WindstresscoefficientEditFieldValueChanged, true);
6830             app.WindstresscoefficientEditField.FontColor = [0.651 0.651 ↙
0.651];
6831             app.WindstresscoefficientEditField.Enable = 'off';
6832             app.WindstresscoefficientEditField.Tooltip = {''};
6833             app.WindstresscoefficientEditField.Position = [518 305 50 18];
6834             app.WindstresscoefficientEditField.Value = 0.002;
6835 
6836             % Create WindstresscoefficientEditFieldLabel
6837             app.WindstresscoefficientEditFieldLabel = uilabel(app.Tab_4);
6838             app.WindstresscoefficientEditFieldLabel.HorizontalAlignment = ↙
'right';
6839             app.WindstresscoefficientEditFieldLabel.Enable = 'off';
6840             app.WindstresscoefficientEditFieldLabel.Tooltip = {'Value used in↙
the quadratic formula'};
6841             app.WindstresscoefficientEditFieldLabel.Position = [341 303 127 ↙
22];
6842             app.WindstresscoefficientEditFieldLabel.Text = 'Wind stress↙
coefficient';
6843 
6844             % Create CrestRatioEditFieldLabel
6845             app.CrestRatioEditFieldLabel = uilabel(app.Tab_4);
6846             app.CrestRatioEditFieldLabel.HorizontalAlignment = 'right';
6847             app.CrestRatioEditFieldLabel.Enable = 'off';
6848             app.CrestRatioEditFieldLabel.Tooltip = {'The ratio of the forced↙
wave crest height to the maximum surface elevation.'; ''; 'For storm surges,↙
default value is 1.'};
6849             app.CrestRatioEditFieldLabel.Position = [341 264 140 22];
6850             app.CrestRatioEditFieldLabel.Text = 'Wind crest percent (ratio)';
6851 
6852             % Create CrestRatioEditField
6853             app.CrestRatioEditField = uieditfield(app.Tab_4, 'numeric');
6854             app.CrestRatioEditField.Limits = [0 Inf];
6855             app.CrestRatioEditField.ValueDisplayFormat = '%11.2f';
6856             app.CrestRatioEditField.ValueChangedFcn = createCallbackFcn(app, ↙
@CrestRatioEditFieldValueChanged, true);
6857             app.CrestRatioEditField.FontColor = [0.651 0.651 0.651];
6858             app.CrestRatioEditField.Enable = 'off';
```

```
6859             app.CrestRatioEditField.Tooltip = {'Ratio of the wave crest height
to the maximum surface height. '; ''; 'Default for storm surge: 1.0'};
6860             app.CrestRatioEditField.Position = [519 266 49 18];
6861             app.CrestRatioEditField.Value = 1;
6862 
6863             % Create ConstantwindfieldSwitchLabel
6864             app.ConstantwindfieldSwitchLabel = uilabel(app.Tab_4);
6865             app.ConstantwindfieldSwitchLabel.HorizontalAlignment = 'center';
6866             app.ConstantwindfieldSwitchLabel.Tooltip = {'Use a constant wind
field'};
6867             app.ConstantwindfieldSwitchLabel.Position = [81 343 109 22];
6868             app.ConstantwindfieldSwitchLabel.Text = 'Constant wind field';
6869 
6870             % Create ConstantwindfieldSwitch
6871             app.ConstantwindfieldSwitch = uiswitch(app.Tab_4, 'slider');
6872             app.ConstantwindfieldSwitch.ValueChangedFcn = createCallbackFcn
(app, @ConstantwindfieldSwitchValueChanged, true);
6873             app.ConstantwindfieldSwitch.Tooltip = {''};
6874             app.ConstantwindfieldSwitch.Position = [243 347 32 15];
6875 
6876             % Create ConstantwindfileEditFieldLabel
6877             app.ConstantwindfileEditFieldLabel = uilabel(app.Tab_4);
6878             app.ConstantwindfileEditFieldLabel.HorizontalAlignment = 'right';
6879             app.ConstantwindfileEditFieldLabel.Enable = 'off';
6880             app.ConstantwindfileEditFieldLabel.Position = [341 344 102 22];
6881             app.ConstantwindfileEditFieldLabel.Text = 'Constant wind file';
6882 
6883             % Create ConstantwindfileEditField
6884             app.ConstantwindfileEditField = uieditfield(app.Tab_4, 'text');
6885             app.ConstantwindfileEditField.Editable = 'off';
6886             app.ConstantwindfileEditField.Enable = 'off';
6887             app.ConstantwindfileEditField.Tooltip = {'Import the wind file'};
6888             app.ConstantwindfileEditField.Position = [453 345 115 20];
6889 
6890             % Create HollandstormmodelSwitchLabel
6891             app.HollandstormmodelSwitchLabel = uilabel(app.Tab_4);
6892             app.HollandstormmodelSwitchLabel.HorizontalAlignment = 'center';
6893             app.HollandstormmodelSwitchLabel.Tooltip = {'Use the Holland
model'};
6894             app.HollandstormmodelSwitchLabel.Position = [81 225 118 22];
6895             app.HollandstormmodelSwitchLabel.Text = 'Holland storm model';
6896 
6897             % Create HollandstormmodelSwitch
6898             app.HollandstormmodelSwitch = uiswitch(app.Tab_4, 'slider');
6899             app.HollandstormmodelSwitch.ValueChangedFcn = createCallbackFcn
(app, @HollandstormmodelSwitchValueChanged, true);
6900             app.HollandstormmodelSwitch.Tooltip = {''};
6901             app.HollandstormmodelSwitch.Position = [243 229 32 14];
6902 
6903             % Create WindwaveinteractionSwitchLabel
6904             app.WindwaveinteractionSwitchLabel = uilabel(app.Tab_4);
6905             app.WindwaveinteractionSwitchLabel.HorizontalAlignment = 'center';
6906             app.WindwaveinteractionSwitchLabel.Enable = 'off';
6907             app.WindwaveinteractionSwitchLabel.Tooltip = {'Activate wind-wave
interaction by Chen et al. (2003)'};
6908             app.WindwaveinteractionSwitchLabel.Position = [81 264 125 22];
6909             app.WindwaveinteractionSwitchLabel.Text = 'Wind-wave interaction';
6910 
```

```
6911            % Create WindwaveinteractionSwitch
6912            app.WindwaveinteractionSwitch = uiswitch(app.Tab_4, 'slider');
6913            app.WindwaveinteractionSwitch.ValueChangedFcn = createCallbackFcn↙
(app, @WindwaveinteractionSwitchValueChanged, true);
6914            app.WindwaveinteractionSwitch.Enable = 'off';
6915            app.WindwaveinteractionSwitch.Tooltip = {''};
6916            app.WindwaveinteractionSwitch.Position = [243 268 32 14];
6917            app.WindwaveinteractionSwitch.Value = 'On';
6918
6919            % Create StormfileEditField
6920            app.StormfileEditField = uieditfield(app.Tab_4, 'text');
6921            app.StormfileEditField.Editable = 'off';
6922            app.StormfileEditField.Enable = 'off';
6923            app.StormfileEditField.Tooltip = {'Import the storm file'};
6924            app.StormfileEditField.Position = [427 226 141 20];
6925
6926            % Create StormfileEditFieldLabel
6927            app.StormfileEditFieldLabel = uilabel(app.Tab_4);
6928            app.StormfileEditFieldLabel.HorizontalAlignment = 'right';
6929            app.StormfileEditFieldLabel.Enable = 'off';
6930            app.StormfileEditFieldLabel.Tooltip = {'It must contain the↙
variables used in the Holland model'};
6931            app.StormfileEditFieldLabel.Position = [341 225 57 22];
6932            app.StormfileEditFieldLabel.Text = 'Storm file';
6933
6934            % Create PressurefileEditFieldLabel
6935            app.PressurefileEditFieldLabel = uilabel(app.Tab_4);
6936            app.PressurefileEditFieldLabel.HorizontalAlignment = 'right';
6937            app.PressurefileEditFieldLabel.Position = [344 429 71 22];
6938            app.PressurefileEditFieldLabel.Text = 'Pressure file';
6939
6940            % Create PressurefileEditField
6941            app.PressurefileEditField = uieditfield(app.Tab_4, 'text');
6942            app.PressurefileEditField.Position = [427 429 141 20];
6943
6944            % Create Button_43
6945            app.Button_43 = uibutton(app.Tab_4, 'push');
6946            app.Button_43.ButtonPushedFcn = createCallbackFcn(app,↙
@Button_43Pushed, true);
6947            app.Button_43.Position = [575 429 22 20];
6948            app.Button_43.Text = '...';
6949
6950            % Create PressureLabel
6951            app.PressureLabel = uilabel(app.Tab_4);
6952            app.PressureLabel.FontWeight = 'bold';
6953            app.PressureLabel.Position = [50 454 56 22];
6954            app.PressureLabel.Text = 'Pressure';
6955
6956            % Create WindLabel
6957            app.WindLabel = uilabel(app.Tab_4);
6958            app.WindLabel.FontWeight = 'bold';
6959            app.WindLabel.Position = [50 370 34 22];
6960            app.WindLabel.Text = 'Wind';
6961
6962            % Create Tab_3
6963            app.Tab_3 = uitab(app.TabGroup2);
6964            app.Tab_3.AutoResizeChildren = 'off';
6965
```

```
6966            % Create Label_6
6967            app.Label_6 = uilabel(app.Tab_3);
6968            app.Label_6.BackgroundColor = [0.651 0.651 0.651];
6969            app.Label_6.Position = [1 62 738 22];
6970            app.Label_6.Text = '';
6971
6972            % Create Label_5
6973            app.Label_5 = uilabel(app.Tab_3);
6974            app.Label_5.BackgroundColor = [0.651 0.651 0.651];
6975            app.Label_5.Position = [1 155 739 22];
6976            app.Label_5.Text = '';
6977
6978            % Create Label_12
6979            app.Label_12 = uilabel(app.Tab_3);
6980            app.Label_12.BackgroundColor = [0.651 0.651 0.651];
6981            app.Label_12.Position = [2 495 738 22];
6982            app.Label_12.Text = '';
6983
6984            % Create SedimentLabel
6985            app.SedimentLabel = uilabel(app.Tab_3);
6986            app.SedimentLabel.FontSize = 14;
6987            app.SedimentLabel.FontWeight = 'bold';
6988            app.SedimentLabel.FontColor = [1 1 1];
6989            app.SedimentLabel.Position = [35 495 69 22];
6990            app.SedimentLabel.Text = 'Sediment';
6991
6992            % Create ButtonGroup_19
6993            app.ButtonGroup_19 = uibuttongroup(app.Tab_3);
6994            app.ButtonGroup_19.AutoResizeChildren = 'off';
6995            app.ButtonGroup_19.SelectionChangedFcn = createCallbackFcn(app, ↙
@ButtonGroup_19SelectionChanged, true);
6996            app.ButtonGroup_19.BorderType = 'none';
6997            app.ButtonGroup_19.Position = [306 459 168 30];
6998
6999            % Create UpwindingButton
7000            app.UpwindingButton = uiradiobutton(app.ButtonGroup_19);
7001            app.UpwindingButton.Text = 'Upwinding';
7002            app.UpwindingButton.Position = [10 6 80 22];
7003            app.UpwindingButton.Value = true;
7004
7005            % Create TVDButton
7006            app.TVDButton = uiradiobutton(app.ButtonGroup_19);
7007            app.TVDButton.Tooltip = {'Total Variation Diminishing '};
7008            app.TVDButton.Text = 'TVD';
7009            app.TVDButton.Position = [102 6 45 22];
7010
7011            % Create NumericalschemeLabel
7012            app.NumericalschemeLabel = uilabel(app.Tab_3);
7013            app.NumericalschemeLabel.Tooltip = {'Scheme to use for solving the↙
advection-diffusion equation'};
7014            app.NumericalschemeLabel.Position = [187 463 105 22];
7015            app.NumericalschemeLabel.Text = 'Numerical scheme';
7016
7017            % Create MediangrainDMEditFieldLabel
7018            app.MediangrainDMEditFieldLabel = uilabel(app.Tab_3);
7019            app.MediangrainDMEditFieldLabel.Tooltip = {'D50 of the grain↙
diameter'};
7020            app.MediangrainDMEditFieldLabel.Position = [187 428 156 22];
```

```
7021             app.MediangrainDMEditFieldLabel.Text = 'Median grain diameter↙
(mm)';
7022
7023             % Create MediangrainDMEditField
7024             app.MediangrainDMEditField = uieditfield(app.Tab_3, 'numeric');
7025             app.MediangrainDMEditField.Limits = [0 Inf];
7026             app.MediangrainDMEditField.ValueDisplayFormat = '%11.3f';
7027             app.MediangrainDMEditField.ValueChangedFcn = createCallbackFcn↙
(app, @MediangrainDMEditFieldValueChanged, true);
7028             app.MediangrainDMEditField.FontColor = [0.651 0.651 0.651];
7029             app.MediangrainDMEditField.Tooltip = {''};
7030             app.MediangrainDMEditField.Position = [490 429 63 19];
7031             app.MediangrainDMEditField.Value = 0.5;
7032
7033             % Create SedimentdensityEditFieldLabel
7034             app.SedimentdensityEditFieldLabel = uilabel(app.Tab_3);
7035             app.SedimentdensityEditFieldLabel.Position = [187 398 99 22];
7036             app.SedimentdensityEditFieldLabel.Text = 'Sediment density';
7037
7038             % Create SedimentdensityEditField
7039             app.SedimentdensityEditField = uieditfield(app.Tab_3, 'numeric');
7040             app.SedimentdensityEditField.Limits = [0 Inf];
7041             app.SedimentdensityEditField.ValueDisplayFormat = '%11.3f';
7042             app.SedimentdensityEditField.ValueChangedFcn = createCallbackFcn↙
(app, @SedimentdensityEditFieldValueChanged, true);
7043             app.SedimentdensityEditField.FontColor = [0.651 0.651 0.651];
7044             app.SedimentdensityEditField.Position = [490 400 63 19];
7045             app.SedimentdensityEditField.Value = 2.68;
7046
7047             % Create SedimentporosityEditFieldLabel
7048             app.SedimentporosityEditFieldLabel = uilabel(app.Tab_3);
7049             app.SedimentporosityEditFieldLabel.Position = [187 369 103 22];
7050             app.SedimentporosityEditFieldLabel.Text = 'Sediment porosity';
7051
7052             % Create SedimentporosityEditField
7053             app.SedimentporosityEditField = uieditfield(app.Tab_3, 'numeric');
7054             app.SedimentporosityEditField.Limits = [0 Inf];
7055             app.SedimentporosityEditField.ValueDisplayFormat = '%11.3f';
7056             app.SedimentporosityEditField.ValueChangedFcn = createCallbackFcn↙
(app, @SedimentporosityEditFieldValueChanged, true);
7057             app.SedimentporosityEditField.FontColor = [0.651 0.651 0.651];
7058             app.SedimentporosityEditField.Position = [490 370 63 19];
7059             app.SedimentporosityEditField.Value = 0.47;
7060
7061             % Create SettlingvelocitymsEditFieldLabel
7062             app.SettlingvelocitymsEditFieldLabel = uilabel(app.Tab_3);
7063             app.SettlingvelocitymsEditFieldLabel.Tooltip = {'Rate at which↙
sediment particles settles'};
7064             app.SettlingvelocitymsEditFieldLabel.Position = [187 340 120 22];
7065             app.SettlingvelocitymsEditFieldLabel.Text = 'Settling velocity↙
(m/s)';
7066
7067             % Create SettlingvelocitymsEditField
7068             app.SettlingvelocitymsEditField = uieditfield(app.Tab_3,↙
'numeric');
7069             app.SettlingvelocitymsEditField.Limits = [0 Inf];
7070             app.SettlingvelocitymsEditField.ValueDisplayFormat = '%11.3f';
7071             app.SettlingvelocitymsEditField.ValueChangedFcn =↙
```

```
createCallbackFcn(app, @SettlingvelocitymsEditFieldValueChanged, true);
7072             app.SettlingvelocitymsEditField.FontColor = [0.651 0.651 0.651];
7073             app.SettlingvelocitymsEditField.Tooltip = {''};
7074             app.SettlingvelocitymsEditField.Position = [490 342 63 19];
7075             app.SettlingvelocitymsEditField.Value = 0.125;
7076
7077             % Create CriticalShieldsEditFieldLabel
7078             app.CriticalShieldsEditFieldLabel = uilabel(app.Tab_3);
7079             app.CriticalShieldsEditFieldLabel.Tooltip = {'Value defining when↙
sediment particles begin to move under the influence of fluid flow'};
7080             app.CriticalShieldsEditFieldLabel.Position = [187 311 241 22];
7081             app.CriticalShieldsEditFieldLabel.Text = 'Critical Shields↙
parameter (suspended load)';
7082
7083             % Create CriticalShieldsEditField
7084             app.CriticalShieldsEditField = uieditfield(app.Tab_3, 'numeric');
7085             app.CriticalShieldsEditField.Limits = [0 Inf];
7086             app.CriticalShieldsEditField.ValueDisplayFormat = '%11.3f';
7087             app.CriticalShieldsEditField.ValueChangedFcn = createCallbackFcn↙
(app, @CriticalShieldsEditFieldValueChanged, true);
7088             app.CriticalShieldsEditField.FontColor = [0.651 0.651 0.651];
7089             app.CriticalShieldsEditField.Tooltip = {''};
7090             app.CriticalShieldsEditField.Position = [490 313 63 19];
7091             app.CriticalShieldsEditField.Value = 0.05;
7092
7093             % Create ShieldsparameterbedloadEditFieldLabel
7094             app.ShieldsparameterbedloadEditFieldLabel = uilabel(app.Tab_3);
7095             app.ShieldsparameterbedloadEditFieldLabel.Tooltip = {'Default↙
value is based on Meyer-Peter and Muller (1984)'};
7096             app.ShieldsparameterbedloadEditFieldLabel.Position = [187 282 156↙
22];
7097             app.ShieldsparameterbedloadEditFieldLabel.Text = 'Shields↙
parameter (bedload)';
7098
7099             % Create ShieldsparamBedloadEditField
7100             app.ShieldsparamBedloadEditField = uieditfield(app.Tab_3,↙
'numeric');
7101             app.ShieldsparamBedloadEditField.Limits = [0 Inf];
7102             app.ShieldsparamBedloadEditField.ValueDisplayFormat = '%11.3f';
7103             app.ShieldsparamBedloadEditField.ValueChangedFcn =↙
createCallbackFcn(app, @ShieldsparamBedloadEditFieldValueChanged, true);
7104             app.ShieldsparamBedloadEditField.FontColor = [0.651 0.651 0.651];
7105             app.ShieldsparamBedloadEditField.Tooltip = {''};
7106             app.ShieldsparamBedloadEditField.Position = [490 284 63 19];
7107             app.ShieldsparamBedloadEditField.Value = 0.047;
7108
7109             % Create RungeKuttaparameter1EditFieldLabel
7110             app.RungeKuttaparameter1EditFieldLabel = uilabel(app.Tab_3);
7111             app.RungeKuttaparameter1EditFieldLabel.Position = [187 253 142↙
22];
7112             app.RungeKuttaparameter1EditFieldLabel.Text = 'Runge-Kutta↙
parameter 1';
7113
7114             % Create RungeKuttaparameter1EditField
7115             app.RungeKuttaparameter1EditField = uieditfield(app.Tab_3,↙
'numeric');
7116             app.RungeKuttaparameter1EditField.Limits = [0 Inf];
7117             app.RungeKuttaparameter1EditField.ValueDisplayFormat = '%11.4f';
```

```
7118             app.RungeKuttaparameter1EditField.ValueChangedFcn = createCallbackFcn(app, @RungeKuttaparameter1EditFieldValueChanged, true);
7119             app.RungeKuttaparameter1EditField.FontColor = [0.651 0.651 0.651];
7120             app.RungeKuttaparameter1EditField.Position = [490 254 63 19];
7121             app.RungeKuttaparameter1EditField.Value = 0.3333;
7122 
7123             % Create RungeKuttaparameter2EditFieldLabel
7124             app.RungeKuttaparameter2EditFieldLabel = uilabel(app.Tab_3);
7125             app.RungeKuttaparameter2EditFieldLabel.Position = [187 224 142 22];
7126             app.RungeKuttaparameter2EditFieldLabel.Text = 'Runge-Kutta parameter 2';
7127 
7128             % Create RungeKuttaparameter2EditField
7129             app.RungeKuttaparameter2EditField = uieditfield(app.Tab_3, 'numeric');
7130             app.RungeKuttaparameter2EditField.Limits = [0 Inf];
7131             app.RungeKuttaparameter2EditField.ValueDisplayFormat = '%11.3f';
7132             app.RungeKuttaparameter2EditField.ValueChangedFcn = createCallbackFcn(app, @RungeKuttaparameter2EditFieldValueChanged, true);
7133             app.RungeKuttaparameter2EditField.FontColor = [0.651 0.651 0.651];
7134             app.RungeKuttaparameter2EditField.Position = [490 226 63 19];
7135             app.RungeKuttaparameter2EditField.Value = 1;
7136 
7137             % Create MinDepthSedPickUpEditFieldLabel
7138             app.MinDepthSedPickUpEditFieldLabel = uilabel(app.Tab_3);
7139             app.MinDepthSedPickUpEditFieldLabel.Position = [187 195 259 22];
7140             app.MinDepthSedPickUpEditFieldLabel.Text = 'Minimum depth for sediment pickup action (m)';
7141 
7142             % Create MinDepthSedPickUpEditField
7143             app.MinDepthSedPickUpEditField = uieditfield(app.Tab_3, 'numeric');
7144             app.MinDepthSedPickUpEditField.Limits = [0 Inf];
7145             app.MinDepthSedPickUpEditField.ValueDisplayFormat = '%11.3f';
7146             app.MinDepthSedPickUpEditField.ValueChangedFcn = createCallbackFcn(app, @MinDepthSedPickUpEditFieldValueChanged, true);
7147             app.MinDepthSedPickUpEditField.FontColor = [0.651 0.651 0.651];
7148             app.MinDepthSedPickUpEditField.Position = [490 197 63 19];
7149             app.MinDepthSedPickUpEditField.Value = 0.1;
7150 
7151             % Create ButtonGroup_20
7152             app.ButtonGroup_20 = uibuttongroup(app.Tab_3);
7153             app.ButtonGroup_20.AutoResizeChildren = 'off';
7154             app.ButtonGroup_20.SelectionChangedFcn = createCallbackFcn(app, @ButtonGroup_20SelectionChanged, true);
7155             app.ButtonGroup_20.BorderType = 'none';
7156             app.ButtonGroup_20.Position = [266 120 241 30];
7157 
7158             % Create UpdatedepthButton
7159             app.UpdatedepthButton = uiradiobutton(app.ButtonGroup_20);
7160             app.UpdatedepthButton.Tooltip = {'Bathymetry is updated in the simulation'};
7161             app.UpdatedepthButton.Text = ' Update depth';
7162             app.UpdatedepthButton.Position = [6 6 99 22];
7163 
7164             % Create NobedchangeButton
7165             app.NobedchangeButton = uiradiobutton(app.ButtonGroup_20);
```

```
7166            app.NobedchangeButton.Tooltip = {'Bathymetry remains constant'};
7167            app.NobedchangeButton.Text = ' No bed change';
7168            app.NobedchangeButton.Position = [121 6 108 22];
7169
7170            % Create Button_36
7171            app.Button_36 = uiradiobutton(app.ButtonGroup_20);
7172            app.Button_36.Visible = 'off';
7173            app.Button_36.Text = '';
7174            app.Button_36.Position = [232 6 25 22];
7175            app.Button_36.Value = true;
7176
7177            % Create BedChangeLabel
7178            app.BedChangeLabel = uilabel(app.Tab_3);
7179            app.BedChangeLabel.Position = [187 125 76 22];
7180            app.BedChangeLabel.Text = 'Bed Change:';
7181
7182            % Create EditFieldLabel
7183            app.EditFieldLabel = uilabel(app.Tab_3);
7184            app.EditFieldLabel.HorizontalAlignment = 'right';
7185            app.EditFieldLabel.Enable = 'off';
7186            app.EditFieldLabel.Visible = 'off';
7187            app.EditFieldLabel.Position = [282 -4 341 22];
7188            app.EditFieldLabel.Text = 'Time step ratio: Boussinesq model to↙
morphological updating';
7189
7190            % Create TimeStepBouss
7191            app.TimeStepBouss = uieditfield(app.Tab_3, 'numeric');
7192            app.TimeStepBouss.Limits = [0 Inf];
7193            app.TimeStepBouss.ValueDisplayFormat = '%11.2f';
7194            app.TimeStepBouss.ValueChangedFcn = createCallbackFcn(app,↙
@TimeStepBoussValueChanged, true);
7195            app.TimeStepBouss.Editable = 'off';
7196            app.TimeStepBouss.FontColor = [0.651 0.651 0.651];
7197            app.TimeStepBouss.Enable = 'off';
7198            app.TimeStepBouss.Visible = 'off';
7199            app.TimeStepBouss.Position = [636 -2 63 19];
7200            app.TimeStepBouss.Value = 25;
7201
7202            % Create AvalancheLabel
7203            app.AvalancheLabel = uilabel(app.Tab_3);
7204            app.AvalancheLabel.FontSize = 14;
7205            app.AvalancheLabel.FontWeight = 'bold';
7206            app.AvalancheLabel.FontColor = [1 1 1];
7207            app.AvalancheLabel.Position = [35 62 74 22];
7208            app.AvalancheLabel.Text = 'Avalanche';
7209
7210            % Create MorphologicalChangeLabel_2
7211            app.MorphologicalChangeLabel_2 = uilabel(app.Tab_3);
7212            app.MorphologicalChangeLabel_2.FontSize = 14;
7213            app.MorphologicalChangeLabel_2.FontWeight = 'bold';
7214            app.MorphologicalChangeLabel_2.FontColor = [1 1 1];
7215            app.MorphologicalChangeLabel_2.Position = [35 155 157 22];
7216            app.MorphologicalChangeLabel_2.Text = 'Morphological Change';
7217
7218            % Create ConsideravalancheLabel
7219            app.ConsideravalancheLabel = uilabel(app.Tab_3);
7220            app.ConsideravalancheLabel.Position = [57 30 118 22];
7221            app.ConsideravalancheLabel.Text = 'Consider avalanche?';
```

```
7222
7223             % Create ButtonGroup_22
7224             app.ButtonGroup_22 = uibuttongroup(app.Tab_3);
7225             app.ButtonGroup_22.AutoResizeChildren = 'off';
7226             app.ButtonGroup_22.SelectionChangedFcn = createCallbackFcn(app, ↙
@ButtonGroup_22SelectionChanged, true);
7227             app.ButtonGroup_22.BorderType = 'none';
7228             app.ButtonGroup_22.Position = [181 27 150 30];
7229
7230             % Create YesButton_8
7231             app.YesButton_8 = uiradiobutton(app.ButtonGroup_22);
7232             app.YesButton_8.Text = 'Yes';
7233             app.YesButton_8.Position = [9 5 41 22];
7234
7235             % Create NoButton_7
7236             app.NoButton_7 = uiradiobutton(app.ButtonGroup_22);
7237             app.NoButton_7.Text = 'No';
7238             app.NoButton_7.Position = [61 5 38 22];
7239             app.NoButton_7.Value = true;
7240
7241             % Create Button_38
7242             app.Button_38 = uiradiobutton(app.ButtonGroup_22);
7243             app.Button_38.Visible = 'off';
7244             app.Button_38.Text = '';
7245             app.Button_38.Position = [103 2 25 22];
7246
7247             % Create RungeKuttaparam2EditFieldLabel_2
7248             app.RungeKuttaparam2EditFieldLabel_2 = uilabel(app.Tab_3);
7249             app.RungeKuttaparam2EditFieldLabel_2.Enable = 'off';
7250             app.RungeKuttaparam2EditFieldLabel_2.Tooltip = {'Steepest angle↙
which the sediments remain stable'; ''; 'Default value 0.7 = 35 deg'};
7251             app.RungeKuttaparam2EditFieldLabel_2.Position = [392 30 154 22];
7252             app.RungeKuttaparam2EditFieldLabel_2.Text = 'Tangent of the repose↙
angle';
7253
7254             % Create RungeKuttaparam2EditField_2
7255             app.RungeKuttaparam2EditField_2 = uieditfield(app.Tab_3, ↙
'numeric');
7256             app.RungeKuttaparam2EditField_2.ValueDisplayFormat = '%11.2f';
7257             app.RungeKuttaparam2EditField_2.ValueChangedFcn = ↙
createCallbackFcn(app, @RungeKuttaparam2EditField_2ValueChanged, true);
7258             app.RungeKuttaparam2EditField_2.FontColor = [0.651 0.651 0.651];
7259             app.RungeKuttaparam2EditField_2.Enable = 'off';
7260             app.RungeKuttaparam2EditField_2.Tooltip = {''};
7261             app.RungeKuttaparam2EditField_2.Position = [558 32 46 19];
7262             app.RungeKuttaparam2EditField_2.Value = 0.7;
7263
7264             % Create FileEditFieldLabel
7265             app.FileEditFieldLabel = uilabel(app.Tab_3);
7266             app.FileEditFieldLabel.HorizontalAlignment = 'right';
7267             app.FileEditFieldLabel.Visible = 'off';
7268             app.FileEditFieldLabel.Position = [558 102 27 22];
7269             app.FileEditFieldLabel.Text = ' File';
7270
7271             % Create FileEditField
7272             app.FileEditField = uieditfield(app.Tab_3, 'text');
7273             app.FileEditField.ValueChangedFcn = createCallbackFcn(app, ↙
@FileEditFieldValueChanged, true);
```

```
7274             app.FileEditField.Editable = 'off';
7275             app.FileEditField.Visible = 'off';
7276             app.FileEditField.Tooltip = {'Import the hard bottom file'; '';↙
'File format and grid dimension must be the same as the input bathymetry'};
7277             app.FileEditField.Position = [592 103 96 20];
7278 
7279             % Create Button_35
7280             app.Button_35 = uibutton(app.Tab_3, 'push');
7281             app.Button_35.ButtonPushedFcn = createCallbackFcn(app,↙
@Button_35Pushed, true);
7282             app.Button_35.Visible = 'off';
7283             app.Button_35.Tooltip = {''};
7284             app.Button_35.Position = [694 103 22 20];
7285             app.Button_35.Text = '...';
7286 
7287             % Create BottomTypeLabel
7288             app.BottomTypeLabel = uilabel(app.Tab_3);
7289             app.BottomTypeLabel.Position = [187 100 77 22];
7290             app.BottomTypeLabel.Text = 'Bottom Type:';
7291 
7292             % Create ButtonGroup_21
7293             app.ButtonGroup_21 = uibuttongroup(app.Tab_3);
7294             app.ButtonGroup_21.AutoResizeChildren = 'off';
7295             app.ButtonGroup_21.SelectionChangedFcn = createCallbackFcn(app,↙
@ButtonGroup_21SelectionChanged, true);
7296             app.ButtonGroup_21.ForegroundColor = [0.651 0.651 0.651];
7297             app.ButtonGroup_21.BorderType = 'none';
7298             app.ButtonGroup_21.Position = [266 96 274 30];
7299 
7300             % Create HardbednoerosionButton
7301             app.HardbednoerosionButton = uiradiobutton(app.ButtonGroup_21);
7302             app.HardbednoerosionButton.Text = ' Hard bed, no erosion';
7303             app.HardbednoerosionButton.Position = [121 5 138 22];
7304 
7305             % Create SedimentbedButton
7306             app.SedimentbedButton = uiradiobutton(app.ButtonGroup_21);
7307             app.SedimentbedButton.Tooltip = {'Allow bed erosion'};
7308             app.SedimentbedButton.Text = ' Sediment bed';
7309             app.SedimentbedButton.Position = [6 5 100 22];
7310 
7311             % Create Button_37
7312             app.Button_37 = uiradiobutton(app.ButtonGroup_21);
7313             app.Button_37.Visible = 'off';
7314             app.Button_37.Text = '';
7315             app.Button_37.Position = [261 4 25 22];
7316             app.Button_37.Value = true;
7317 
7318             % Create TimeStepSedToMorpEditFieldLabel
7319             app.TimeStepSedToMorpEditFieldLabel = uilabel(app.Tab_3);
7320             app.TimeStepSedToMorpEditFieldLabel.HorizontalAlignment = 'right';
7321             app.TimeStepSedToMorpEditFieldLabel.Enable = 'off';
7322             app.TimeStepSedToMorpEditFieldLabel.Visible = 'off';
7323             app.TimeStepSedToMorpEditFieldLabel.Position = [331 194 237 22];
7324             app.TimeStepSedToMorpEditFieldLabel.Text = 'Time step ratio:↙
sediment to morphological';
7325 
7326             % Create TimeStepSedToMorphlEditField
7327             app.TimeStepSedToMorphlEditField = uieditfield(app.Tab_3,↙
```

```
'numeric');
7328            app.TimeStepSedToMorphlEditField.Limits = [0 Inf];
7329            app.TimeStepSedToMorphlEditField.ValueChangedFcn = ↙
createCallbackFcn(app, @TimeStepSedToMorphlEditFieldValueChanged, true);
7330            app.TimeStepSedToMorphlEditField.Editable = 'off';
7331            app.TimeStepSedToMorphlEditField.FontColor = [0.651 0.651 0.651];
7332            app.TimeStepSedToMorphlEditField.Enable = 'off';
7333            app.TimeStepSedToMorphlEditField.Visible = 'off';
7334            app.TimeStepSedToMorphlEditField.Position = [491 187 63 19];
7335            app.TimeStepSedToMorphlEditField.Value = 10;
7336
7337            % Create AddrollereffectsCheckBox
7338            app.AddrollereffectsCheckBox = uicheckbox(app.Tab_3);
7339            app.AddrollereffectsCheckBox.Text = 'Add roller effects';
7340            app.AddrollereffectsCheckBox.Position = [490 463 113 22];
7341
7342            % Create Tab_6
7343            app.Tab_6 = uitab(app.TabGroup2);
7344            app.Tab_6.AutoResizeChildren = 'off';
7345
7346            % Create Label_13
7347            app.Label_13 = uilabel(app.Tab_6);
7348            app.Label_13.BackgroundColor = [0.651 0.651 0.651];
7349            app.Label_13.Position = [1 495 738 22];
7350            app.Label_13.Text = '';
7351
7352            % Create ImportVesselFilesLabel
7353            app.ImportVesselFilesLabel = uilabel(app.Tab_6);
7354            app.ImportVesselFilesLabel.Tooltip = {''};
7355            app.ImportVesselFilesLabel.Position = [77 452 106 22];
7356            app.ImportVesselFilesLabel.Text = 'Import vessel files:';
7357
7358            % Create ImportvesselfilesTextArea
7359            app.ImportvesselfilesTextArea = uitextarea(app.Tab_6);
7360            app.ImportvesselfilesTextArea.Editable = 'off';
7361            app.ImportvesselfilesTextArea.Position = [77 324 154 120];
7362
7363            % Create Button_40
7364            app.Button_40 = uibutton(app.Tab_6, 'push');
7365            app.Button_40.ButtonPushedFcn = createCallbackFcn(app, ↙
@Button_40Pushed, true);
7366            app.Button_40.Tooltip = {'Find vessel files'};
7367            app.Button_40.Position = [240 374 26 20];
7368            app.Button_40.Text = '...';
7369
7370            % Create VesselLabel
7371            app.VesselLabel = uilabel(app.Tab_6);
7372            app.VesselLabel.FontSize = 14;
7373            app.VesselLabel.FontWeight = 'bold';
7374            app.VesselLabel.FontColor = [1 1 1];
7375            app.VesselLabel.Position = [24 495 48 22];
7376            app.VesselLabel.Text = 'Vessel';
7377
7378            % Create Label_8
7379            app.Label_8 = uilabel(app.Tab_6);
7380            app.Label_8.BackgroundColor = [0.651 0.651 0.651];
7381            app.Label_8.Position = [1 197 738 22];
7382            app.Label_8.Text = '';
```

```
7383
7384            % Create DeepDraftVesselsLabel
7385            app.DeepDraftVesselsLabel = uilabel(app.Tab_6);
7386            app.DeepDraftVesselsLabel.FontWeight = 'bold';
7387            app.DeepDraftVesselsLabel.FontColor = [1 1 1];
7388            app.DeepDraftVesselsLabel.Position = [24 197 114 22];
7389            app.DeepDraftVesselsLabel.Text = 'Deep Draft Vessels';
7390
7391            % Create MinimumClearanceLabel
7392            app.MinimumClearanceLabel = uilabel(app.Tab_6);
7393            app.MinimumClearanceLabel.Enable = 'off';
7394            app.MinimumClearanceLabel.Position = [80 132 116 22];
7395            app.MinimumClearanceLabel.Text = 'Minimum Clearance:';
7396
7397            % Create MinimumClearanceEditField
7398            app.MinimumClearanceEditField = uieditfield(app.Tab_6, 'numeric');
7399            app.MinimumClearanceEditField.Limits = [0 Inf];
7400            app.MinimumClearanceEditField.ValueDisplayFormat = '%5.1f';
7401            app.MinimumClearanceEditField.ValueChangedFcn = createCallbackFcn↙
(app, @MinimumClearanceEditFieldValueChanged, true);
7402            app.MinimumClearanceEditField.FontColor = [0.651 0.651 0.651];
7403            app.MinimumClearanceEditField.Enable = 'off';
7404            app.MinimumClearanceEditField.Position = [198 134 35 19];
7405            app.MinimumClearanceEditField.Value = 1;
7406
7407            % Create IncludeSedimentEffectintheModelCheckBox
7408            app.IncludeSedimentEffectintheModelCheckBox = uicheckbox(app.↙
Tab_6);
7409            app.IncludeSedimentEffectintheModelCheckBox.Text = 'Include↙
Sediment Effect in the Model';
7410            app.IncludeSedimentEffectintheModelCheckBox.Position = [75 296 221↙
22];
7411
7412            % Create ActivateCheckBox
7413            app.ActivateCheckBox = uicheckbox(app.Tab_6);
7414            app.ActivateCheckBox.ValueChangedFcn = createCallbackFcn(app,↙
@ActivateCheckBoxValueChanged, true);
7415            app.ActivateCheckBox.Tooltip = {'Check the box to address the↙
instability issue that may arise when modelling a large vessel with a draft close↙
to the channel depth'};
7416            app.ActivateCheckBox.Text = '  Activate to reduce high-frequency↙
spikes occurring close to the ship';
7417            app.ActivateCheckBox.Position = [81 163 401 22];
7418
7419            % Create ShockcapturingCheckBox
7420            app.ShockcapturingCheckBox = uicheckbox(app.Tab_6);
7421            app.ShockcapturingCheckBox.Enable = 'off';
7422            app.ShockcapturingCheckBox.Tooltip = {'Utilizes shock-capturing↙
method'};
7423            app.ShockcapturingCheckBox.Text = 'Shock capturing';
7424            app.ShockcapturingCheckBox.Position = [451 72 110 22];
7425            app.ShockcapturingCheckBox.Value = true;
7426
7427            % Create FrictionCheckBox_2
7428            app.FrictionCheckBox_2 = uicheckbox(app.Tab_6);
7429            app.FrictionCheckBox_2.ValueChangedFcn = createCallbackFcn(app,↙
@FrictionCheckBox_2ValueChanged, true);
7430            app.FrictionCheckBox 2.Enable = 'off';
```

```
7431            app.FrictionCheckBox_2.Tooltip = {'Utilizes dissipation method for damping waves similar to a sponge layer'};
7432            app.FrictionCheckBox_2.Text = 'Friction';
7433            app.FrictionCheckBox_2.Position = [80 72 62 22];
7434            app.FrictionCheckBox_2.Value = true;
7435
7436            % Create ViscosityCheckBox
7437            app.ViscosityCheckBox = uicheckbox(app.Tab_6);
7438            app.ViscosityCheckBox.ValueChangedFcn = createCallbackFcn(app, @ViscosityCheckBoxValueChanged, true);
7439            app.ViscosityCheckBox.Enable = 'off';
7440            app.ViscosityCheckBox.Tooltip = {'Similar to friction method but its wave damping rate is usually lower'};
7441            app.ViscosityCheckBox.Text = 'Viscosity';
7442            app.ViscosityCheckBox.Position = [279 72 70 22];
7443
7444            % Create FrictionCoefficientEditFieldLabel
7445            app.FrictionCoefficientEditFieldLabel = uilabel(app.Tab_6);
7446            app.FrictionCoefficientEditFieldLabel.HorizontalAlignment = 'right';
7447            app.FrictionCoefficientEditFieldLabel.Enable = 'off';
7448            app.FrictionCoefficientEditFieldLabel.Tooltip = {'Suggested values: 0.1 - 1'};
7449            app.FrictionCoefficientEditFieldLabel.Position = [136 74 63 19];
7450            app.FrictionCoefficientEditFieldLabel.Text = 'Coefficient';
7451
7452            % Create FrictionCoefficientEditField
7453            app.FrictionCoefficientEditField = uieditfield(app.Tab_6, 'numeric');
7454            app.FrictionCoefficientEditField.Limits = [0 Inf];
7455            app.FrictionCoefficientEditField.ValueDisplayFormat = '%5.1f';
7456            app.FrictionCoefficientEditField.ValueChangedFcn = createCallbackFcn(app, @FrictionCoefficientEditFieldValueChanged, true);
7457            app.FrictionCoefficientEditField.FontColor = [0.651 0.651 0.651];
7458            app.FrictionCoefficientEditField.Enable = 'off';
7459            app.FrictionCoefficientEditField.Tooltip = {''};
7460            app.FrictionCoefficientEditField.Position = [209 74 35 19];
7461            app.FrictionCoefficientEditField.Value = 0.1;
7462
7463            % Create ValueEditFieldLabel_2
7464            app.ValueEditFieldLabel_2 = uilabel(app.Tab_6);
7465            app.ValueEditFieldLabel_2.HorizontalAlignment = 'right';
7466            app.ValueEditFieldLabel_2.Enable = 'off';
7467            app.ValueEditFieldLabel_2.Tooltip = {'Suggested values: 0.1 - 5'};
7468            app.ValueEditFieldLabel_2.Position = [343 74 34 19];
7469            app.ValueEditFieldLabel_2.Text = 'Value';
7470
7471            % Create ValueEditField
7472            app.ValueEditField = uieditfield(app.Tab_6, 'numeric');
7473            app.ValueEditField.Limits = [0 Inf];
7474            app.ValueEditField.ValueDisplayFormat = '%5.1f';
7475            app.ValueEditField.ValueChangedFcn = createCallbackFcn(app, @ValueEditFieldValueChanged, true);
7476            app.ValueEditField.Editable = 'off';
7477            app.ValueEditField.FontColor = [0.651 0.651 0.651];
7478            app.ValueEditField.Enable = 'off';
7479            app.ValueEditField.Tooltip = {''};
7480            app.ValueEditField.Position = [384 74 35 19];
```

```
7481            app.ValueEditField.Value = 1;
7482
7483            % Create SelectMethodLabel
7484            app.SelectMethodLabel = uilabel(app.Tab_6);
7485            app.SelectMethodLabel.FontWeight = 'bold';
7486            app.SelectMethodLabel.Enable = 'off';
7487            app.SelectMethodLabel.Position = [81 97 92 22];
7488            app.SelectMethodLabel.Text = 'Select Method:';
7489
7490            % Create SelectInitialConditonLabel
7491            app.SelectInitialConditonLabel = uilabel(app.InitialConditionTab);
7492            app.SelectInitialConditonLabel.FontWeight = 'bold';
7493            app.SelectInitialConditonLabel.Position = [21 614 135 22];
7494            app.SelectInitialConditonLabel.Text = 'Select Initial Conditon:';
7495
7496            % Create ButtonGroup_18
7497            app.ButtonGroup_18 = uibuttongroup(app.InitialConditionTab);
7498            app.ButtonGroup_18.AutoResizeChildren = 'off';
7499            app.ButtonGroup_18.SelectionChangedFcn = createCallbackFcn(app, ↙
@ButtonGroup_6SelectionChanged, true);
7500            app.ButtonGroup_18.BorderType = 'none';
7501            app.ButtonGroup_18.FontSize = 14;
7502            app.ButtonGroup_18.Position = [167 610 412 30];
7503
7504            % Create ImportUVZButton
7505            app.ImportUVZButton = uiradiobutton(app.ButtonGroup_18);
7506            app.ImportUVZButton.Text = 'Import UVZ';
7507            app.ImportUVZButton.Position = [3 5 84 22];
7508
7509            % Create WavemakerButton
7510            app.WavemakerButton = uiradiobutton(app.ButtonGroup_18);
7511            app.WavemakerButton.Text = 'Wavemaker';
7512            app.WavemakerButton.Position = [109 5 85 22];
7513
7514            % Create Button_31
7515            app.Button_31 = uiradiobutton(app.ButtonGroup_18);
7516            app.Button_31.Enable = 'off';
7517            app.Button_31.Visible = 'off';
7518            app.Button_31.Text = '';
7519            app.Button_31.Position = [395 5 25 22];
7520            app.Button_31.Value = true;
7521
7522            % Create MeteotsunamiButton
7523            app.MeteotsunamiButton = uiradiobutton(app.ButtonGroup_18);
7524            app.MeteotsunamiButton.Tooltip = {'Initiate the tsunami using↙
atmospheric disturbances'};
7525            app.MeteotsunamiButton.Text = 'Meteotsunami';
7526            app.MeteotsunamiButton.Position = [215 5 98 22];
7527
7528            % Create VesselButton
7529            app.VesselButton = uiradiobutton(app.ButtonGroup_18);
7530            app.VesselButton.Tooltip = {'Use the ship-wake model in the↙
simulation'};
7531            app.VesselButton.Text = 'Vessel';
7532            app.VesselButton.Position = [334 5 56 22];
7533
7534            % Create GenerateinputtxtButton_4
7535            app.GenerateinputtxtButton_4 = uibutton(app.InitialConditionTab, ↙
```

```
'push');
7536            app.GenerateinputtxtButton_4.ButtonPushedFcn = createCallbackFcn↙
(app, @GenerateinputtxtButton_4Pushed, true);
7537            app.GenerateinputtxtButton_4.HandleVisibility = 'off';
7538            app.GenerateinputtxtButton_4.BackgroundColor = [0.9216 0.9216↙
0.9216];
7539            app.GenerateinputtxtButton_4.FontSize = 15;
7540            app.GenerateinputtxtButton_4.FontWeight = 'bold';
7541            app.GenerateinputtxtButton_4.Position = [186 23 355 43];
7542            app.GenerateinputtxtButton_4.Text = 'Generate input.txt';
7543
7544            % Create SedimentTransportCheckBox
7545            app.SedimentTransportCheckBox = uicheckbox(app.↙
InitialConditionTab);
7546            app.SedimentTransportCheckBox.ValueChangedFcn = createCallbackFcn↙
(app, @SedimentTransportCheckBoxValueChanged, true);
7547            app.SedimentTransportCheckBox.Text = 'Sediment Transport';
7548            app.SedimentTransportCheckBox.Position = [584 614 127 22];
7549
7550            % Create ExpectedOutputTab
7551            app.ExpectedOutputTab = uitab(app.TabGroup);
7552            app.ExpectedOutputTab.AutoResizeChildren = 'off';
7553            app.ExpectedOutputTab.Tooltip = {'Flux F in X direction'};
7554            app.ExpectedOutputTab.Title = 'Expected Output';
7555            app.ExpectedOutputTab.BackgroundColor = [0.9412 0.9412 0.9412];
7556            app.ExpectedOutputTab.Scrollable = 'on';
7557
7558            % Create GridLayout
7559            app.GridLayout = uigridlayout(app.ExpectedOutputTab);
7560            app.GridLayout.ColumnWidth = {'1x', 31, 45, 36, 39, 38, 54, 67,↙
66, 76, 57, '2.84x'};
7561            app.GridLayout.RowHeight = {22, 22, 22, 22, 22, 22, '1x', 22, 22,↙
22, 22, 22, 22, 22, 22, 22, '1.75x', 43};
7562
7563            % Create DepthCheckBox
7564            app.DepthCheckBox = uicheckbox(app.GridLayout);
7565            app.DepthCheckBox.ValueChangedFcn = createCallbackFcn(app,↙
@DepthCheckBoxValueChanged, true);
7566            app.DepthCheckBox.Tooltip = {'DEPTH_OUT'};
7567            app.DepthCheckBox.Text = 'Depth';
7568            app.DepthCheckBox.Layout.Row = 6;
7569            app.DepthCheckBox.Layout.Column = 7;
7570
7571            % Create MaxVorticityCheckBox
7572            app.MaxVorticityCheckBox = uicheckbox(app.GridLayout);
7573            app.MaxVorticityCheckBox.ValueChangedFcn = createCallbackFcn(app,↙
@MaxVorticityCheckBoxValueChanged, true);
7574            app.MaxVorticityCheckBox.Tooltip = {'VORMAX'};
7575            app.MaxVorticityCheckBox.Text = 'Maximum Vorticity';
7576            app.MaxVorticityCheckBox.Layout.Row = 9;
7577            app.MaxVorticityCheckBox.Layout.Column = [7 8];
7578
7579            % Create UVelocityCheckBox
7580            app.UVelocityCheckBox = uicheckbox(app.GridLayout);
7581            app.UVelocityCheckBox.ValueChangedFcn = createCallbackFcn(app,↙
@UVelocityCheckBoxValueChanged, true);
7582            app.UVelocityCheckBox.Tooltip = {'U: Velocity in X direction'};
7583            app.UVelocityCheckBox.Text = 'U Velocity';
```

```
7584            app.UVelocityCheckBox.Layout.Row = 2;
7585            app.UVelocityCheckBox.Layout.Column = 10;
7586
7587            % Create MaxMomentumFluxCheckBox
7588            app.MaxMomentumFluxCheckBox = uicheckbox(app.GridLayout);
7589            app.MaxMomentumFluxCheckBox.ValueChangedFcn = createCallbackFcn↙
(app, @MaxMomentumFluxCheckBoxValueChanged, true);
7590            app.MaxMomentumFluxCheckBox.Tooltip = {'MFMAX'};
7591            app.MaxMomentumFluxCheckBox.Text = 'Maximum Momentum Flux';
7592            app.MaxMomentumFluxCheckBox.Layout.Row = 11;
7593            app.MaxMomentumFluxCheckBox.Layout.Column = [2 5];
7594
7595            % Create VVelocityCheckBox
7596            app.VVelocityCheckBox = uicheckbox(app.GridLayout);
7597            app.VVelocityCheckBox.ValueChangedFcn = createCallbackFcn(app,↙
@VVelocityCheckBoxValueChanged, true);
7598            app.VVelocityCheckBox.Tooltip = {' in y direction'};
7599            app.VVelocityCheckBox.Text = 'V Velocity';
7600            app.VVelocityCheckBox.Layout.Row = 3;
7601            app.VVelocityCheckBox.Layout.Column = 10;
7602
7603            % Create ArrivalCheckBox
7604            app.ArrivalCheckBox = uicheckbox(app.GridLayout);
7605            app.ArrivalCheckBox.ValueChangedFcn = createCallbackFcn(app,↙
@ArrivalCheckBoxValueChanged, true);
7606            app.ArrivalCheckBox.Tooltip = {'OUT_Time'};
7607            app.ArrivalCheckBox.Text = 'Arrival';
7608            app.ArrivalCheckBox.Layout.Row = 2;
7609            app.ArrivalCheckBox.Layout.Column = [7 8];
7610
7611            % Create SurfaceElevationattimetCheckBox
7612            app.SurfaceElevationattimetCheckBox = uicheckbox(app.GridLayout);
7613            app.SurfaceElevationattimetCheckBox.ValueChangedFcn =↙
createCallbackFcn(app, @SurfaceElevationattimetCheckBoxValueChanged, true);
7614            app.SurfaceElevationattimetCheckBox.Tooltip = {'ETA: Surface↙
elevation at time t'};
7615            app.SurfaceElevationattimetCheckBox.Text = 'Surface Elevation at↙
time t';
7616            app.SurfaceElevationattimetCheckBox.Layout.Row = 4;
7617            app.SurfaceElevationattimetCheckBox.Layout.Column = [2 5];
7618            app.SurfaceElevationattimetCheckBox.Value = true;
7619
7620            % Create WetdrymaskCheckBox
7621            app.WetdrymaskCheckBox = uicheckbox(app.GridLayout);
7622            app.WetdrymaskCheckBox.ValueChangedFcn = createCallbackFcn(app,↙
@WetdrymaskCheckBoxValueChanged, true);
7623            app.WetdrymaskCheckBox.Tooltip = {'MASK'};
7624            app.WetdrymaskCheckBox.Text = 'Wet-dry mask';
7625            app.WetdrymaskCheckBox.Layout.Row = 9;
7626            app.WetdrymaskCheckBox.Layout.Column = [10 11];
7627
7628            % Create WaveHeightCheckBox
7629            app.WaveHeightCheckBox = uicheckbox(app.GridLayout);
7630            app.WaveHeightCheckBox.ValueChangedFcn = createCallbackFcn(app,↙
@WaveHeightCheckBoxValueChanged, true);
7631            app.WaveHeightCheckBox.Tooltip = {'Wave height: Hsig,Hrms, Havg'};
7632            app.WaveHeightCheckBox.Text = 'Wave Height';
7633            app.WaveHeightCheckBox.Layout.Row = 5;
```

```
7634            app.WaveHeightCheckBox.Layout.Column = [2 3];
7635
7636            % Create WetdrymaskforBoussinesqNSWECheckBox
7637            app.WetdrymaskforBoussinesqNSWECheckBox = uicheckbox(app.↙
GridLayout);
7638            app.WetdrymaskforBoussinesqNSWECheckBox.ValueChangedFcn =↙
createCallbackFcn(app, @WetdrymaskforBoussinesqNSWECheckBoxValueChanged, true);
7639            app.WetdrymaskforBoussinesqNSWECheckBox.Tooltip = {'MASK9: Crate a↙
wetting-drying mask for Boussinesq/NSWE modelling'};
7640            app.WetdrymaskforBoussinesqNSWECheckBox.Text = 'Wet-dry mask for↙
Boussinesq/NSWE';
7641            app.WetdrymaskforBoussinesqNSWECheckBox.Layout.Row = 10;
7642            app.WetdrymaskforBoussinesqNSWECheckBox.Layout.Column = [10 12];
7643
7644            % Create RollerinducedFluxCheckBox
7645            app.RollerinducedFluxCheckBox = uicheckbox(app.GridLayout);
7646            app.RollerinducedFluxCheckBox.ValueChangedFcn = createCallbackFcn↙
(app, @RollerinducedFluxCheckBoxValueChanged, true);
7647            app.RollerinducedFluxCheckBox.Tooltip = {'ROLLER: Roller-induced↙
mass flux'};
7648            app.RollerinducedFluxCheckBox.Text = 'Roller-induced Flux';
7649            app.RollerinducedFluxCheckBox.Layout.Row = 16;
7650            app.RollerinducedFluxCheckBox.Layout.Column = [2 4];
7651
7652            % Create XSourceCheckBox
7653            app.XSourceCheckBox = uicheckbox(app.GridLayout);
7654            app.XSourceCheckBox.ValueChangedFcn = createCallbackFcn(app, ↙
@XSourceCheckBoxValueChanged, true);
7655            app.XSourceCheckBox.Tooltip = {'SoureceX: Source terms in X↙
direction'};
7656            app.XSourceCheckBox.Text = 'X Source';
7657            app.XSourceCheckBox.Layout.Row = 4;
7658            app.XSourceCheckBox.Layout.Column = [7 8];
7659
7660            % Create PressureFieldCheckBox
7661            app.PressureFieldCheckBox = uicheckbox(app.GridLayout);
7662            app.PressureFieldCheckBox.ValueChangedFcn = createCallbackFcn(app, ↙
@PressureFieldCheckBoxValueChanged, true);
7663            app.PressureFieldCheckBox.Tooltip = {'OUT_METEO'};
7664            app.PressureFieldCheckBox.Text = 'Pressure Field';
7665            app.PressureFieldCheckBox.Layout.Row = 13;
7666            app.PressureFieldCheckBox.Layout.Column = [7 8];
7667
7668            % Create YSourceCheckBox
7669            app.YSourceCheckBox = uicheckbox(app.GridLayout);
7670            app.YSourceCheckBox.ValueChangedFcn = createCallbackFcn(app, ↙
@YSourceCheckBoxValueChanged, true);
7671            app.YSourceCheckBox.Tooltip = {'SoureceY: Source terms in Y↙
direction'};
7672            app.YSourceCheckBox.Text = 'Y Source';
7673            app.YSourceCheckBox.Layout.Row = 5;
7674            app.YSourceCheckBox.Layout.Column = [7 8];
7675
7676            % Create BreakingLocationCheckBox
7677            app.BreakingLocationCheckBox = uicheckbox(app.GridLayout);
7678            app.BreakingLocationCheckBox.ValueChangedFcn = createCallbackFcn↙
(app, @BreakingLocationCheckBoxValueChanged, true);
7679            app.BreakingLocationCheckBox.Tooltip = {'OUT NU'};
```

```
7680            app.BreakingLocationCheckBox.Text = 'Breaking Location';
7681            app.BreakingLocationCheckBox.Layout.Row = 11;
7682            app.BreakingLocationCheckBox.Layout.Column = [7 8];
7683
7684            % Create PMomentumFluxCheckBox
7685            app.PMomentumFluxCheckBox = uicheckbox(app.GridLayout);
7686            app.PMomentumFluxCheckBox.ValueChangedFcn = createCallbackFcn(app,
@PMomentumFluxCheckBoxValueChanged, true);
7687            app.PMomentumFluxCheckBox.Tooltip = {'P: Momentum flux in X
direction'};
7688            app.PMomentumFluxCheckBox.Text = 'P Momentum Flux';
7689            app.PMomentumFluxCheckBox.Layout.Row = 9;
7690            app.PMomentumFluxCheckBox.Layout.Column = [2 4];
7691
7692            % Create QmomentumFluxCheckBox
7693            app.QmomentumFluxCheckBox = uicheckbox(app.GridLayout);
7694            app.QmomentumFluxCheckBox.ValueChangedFcn = createCallbackFcn(app,
@QmomentumFluxCheckBoxValueChanged, true);
7695            app.QmomentumFluxCheckBox.Tooltip = {'Q: Momentum flux in Y
direction'};
7696            app.QmomentumFluxCheckBox.Text = 'Q momentum Flux';
7697            app.QmomentumFluxCheckBox.Layout.Row = 10;
7698            app.QmomentumFluxCheckBox.Layout.Column = [2 4];
7699
7700            % Create FxFluxCheckBox
7701            app.FxFluxCheckBox = uicheckbox(app.GridLayout);
7702            app.FxFluxCheckBox.ValueChangedFcn = createCallbackFcn(app,
@FxFluxCheckBoxValueChanged, true);
7703            app.FxFluxCheckBox.Tooltip = {'Flux F in X direction'};
7704            app.FxFluxCheckBox.Text = 'Fx Flux';
7705            app.FxFluxCheckBox.Layout.Row = 12;
7706            app.FxFluxCheckBox.Layout.Column = [2 3];
7707
7708            % Create FyFluxCheckBox
7709            app.FyFluxCheckBox = uicheckbox(app.GridLayout);
7710            app.FyFluxCheckBox.ValueChangedFcn = createCallbackFcn(app,
@FyFluxCheckBoxValueChanged, true);
7711            app.FyFluxCheckBox.Tooltip = {'Flux F in Y direction'};
7712            app.FyFluxCheckBox.Text = 'Fy Flux';
7713            app.FyFluxCheckBox.Layout.Row = 13;
7714            app.FyFluxCheckBox.Layout.Column = [2 3];
7715
7716            % Create GxFluxCheckBox
7717            app.GxFluxCheckBox = uicheckbox(app.GridLayout);
7718            app.GxFluxCheckBox.ValueChangedFcn = createCallbackFcn(app,
@GxFluxCheckBoxValueChanged, true);
7719            app.GxFluxCheckBox.Tooltip = {'Gx: Flux G in X direction'};
7720            app.GxFluxCheckBox.Text = 'Gx Flux';
7721            app.GxFluxCheckBox.Layout.Row = 14;
7722            app.GxFluxCheckBox.Layout.Column = [2 3];
7723
7724            % Create GyFluxCheckBox
7725            app.GyFluxCheckBox = uicheckbox(app.GridLayout);
7726            app.GyFluxCheckBox.ValueChangedFcn = createCallbackFcn(app,
@GyFluxCheckBoxValueChanged, true);
7727            app.GyFluxCheckBox.Tooltip = {'Gy: Flux G in Y direction'};
7728            app.GyFluxCheckBox.Text = 'Gy Flux';
7729            app.GyFluxCheckBox.Layout.Row = 15;
```

```
7730            app.GyFluxCheckBox.Layout.Column = [2 3];
7731
7732            % Create BreakingAgeCheckBox
7733            app.BreakingAgeCheckBox = uicheckbox(app.GridLayout);
7734            app.BreakingAgeCheckBox.ValueChangedFcn = createCallbackFcn(app, ↙
@BreakingAgeCheckBoxValueChanged, true);
7735            app.BreakingAgeCheckBox.Tooltip = {'AGE: Breaking Age'};
7736            app.BreakingAgeCheckBox.Text = 'Breaking Age';
7737            app.BreakingAgeCheckBox.Layout.Row = 10;
7738            app.BreakingAgeCheckBox.Layout.Column = [7 8];
7739
7740            % Create MaximumVelocityCheckBox
7741            app.MaximumVelocityCheckBox = uicheckbox(app.GridLayout);
7742            app.MaximumVelocityCheckBox.ValueChangedFcn = createCallbackFcn ↙
(app, @MaximumVelocityCheckBoxValueChanged, true);
7743            app.MaximumVelocityCheckBox.Tooltip = {'Umax: Maximum velocity'};
7744            app.MaximumVelocityCheckBox.Text = 'Maximum Velocity ';
7745            app.MaximumVelocityCheckBox.Layout.Row = 6;
7746            app.MaximumVelocityCheckBox.Layout.Column = [10 11];
7747
7748            % Create UndertowCheckBox
7749            app.UndertowCheckBox = uicheckbox(app.GridLayout);
7750            app.UndertowCheckBox.ValueChangedFcn = createCallbackFcn(app, ↙
@UndertowCheckBoxValueChanged, true);
7751            app.UndertowCheckBox.Tooltip = {'UNDERTOW: Roller-induced extra ↙
undertow flux'};
7752            app.UndertowCheckBox.Text = 'Undertow';
7753            app.UndertowCheckBox.Layout.Row = 12;
7754            app.UndertowCheckBox.Layout.Column = [7 8];
7755
7756            % Create VelocityLabel
7757            app.VelocityLabel = uilabel(app.GridLayout);
7758            app.VelocityLabel.FontSize = 13;
7759            app.VelocityLabel.FontWeight = 'bold';
7760            app.VelocityLabel.FontColor = [0.0314 0.3686 0.6];
7761            app.VelocityLabel.Layout.Row = 1;
7762            app.VelocityLabel.Layout.Column = 10;
7763            app.VelocityLabel.Text = 'Velocity';
7764
7765            % Create WaveHeightLabel
7766            app.WaveHeightLabel = uilabel(app.GridLayout);
7767            app.WaveHeightLabel.FontSize = 13;
7768            app.WaveHeightLabel.FontWeight = 'bold';
7769            app.WaveHeightLabel.FontColor = [0.0314 0.3686 0.6];
7770            app.WaveHeightLabel.Layout.Row = 1;
7771            app.WaveHeightLabel.Layout.Column = [2 3];
7772            app.WaveHeightLabel.Text = 'Wave Height';
7773
7774            % Create SourceLabel
7775            app.SourceLabel = uilabel(app.GridLayout);
7776            app.SourceLabel.FontSize = 13;
7777            app.SourceLabel.FontWeight = 'bold';
7778            app.SourceLabel.FontColor = [0.0314 0.3686 0.6];
7779            app.SourceLabel.Layout.Row = 3;
7780            app.SourceLabel.Layout.Column = 7;
7781            app.SourceLabel.Text = 'Source';
7782
7783            % Create MaskLabel
```

```
7784            app.MaskLabel = uilabel(app.GridLayout);
7785            app.MaskLabel.FontSize = 13;
7786            app.MaskLabel.FontWeight = 'bold';
7787            app.MaskLabel.FontColor = [0.0314 0.3686 0.6];
7788            app.MaskLabel.Layout.Row = 8;
7789            app.MaskLabel.Layout.Column = 10;
7790            app.MaskLabel.Text = 'Mask';
7791
7792            % Create FluxLabel
7793            app.FluxLabel = uilabel(app.GridLayout);
7794            app.FluxLabel.FontSize = 13;
7795            app.FluxLabel.FontWeight = 'bold';
7796            app.FluxLabel.FontColor = [0.0314 0.3686 0.6];
7797            app.FluxLabel.Layout.Row = 8;
7798            app.FluxLabel.Layout.Column = 2;
7799            app.FluxLabel.Text = 'Flux';
7800
7801            % Create TimeLabel
7802            app.TimeLabel = uilabel(app.GridLayout);
7803            app.TimeLabel.FontSize = 13;
7804            app.TimeLabel.FontWeight = 'bold';
7805            app.TimeLabel.FontColor = [0.0314 0.3686 0.6];
7806            app.TimeLabel.Layout.Row = 1;
7807            app.TimeLabel.Layout.Column = 7;
7808            app.TimeLabel.Text = 'Time';
7809
7810            % Create OthersLabel
7811            app.OthersLabel = uilabel(app.GridLayout);
7812            app.OthersLabel.FontSize = 13;
7813            app.OthersLabel.FontWeight = 'bold';
7814            app.OthersLabel.FontColor = [0.0314 0.3686 0.6];
7815            app.OthersLabel.Layout.Row = 8;
7816            app.OthersLabel.Layout.Column = 7;
7817            app.OthersLabel.Text = 'Others';
7818
7819            % Create GenerateinputtxtButton_2
7820            app.GenerateinputtxtButton_2 = uibutton(app.GridLayout, 'push');
7821            app.GenerateinputtxtButton_2.ButtonPushedFcn = createCallbackFcn↙
(app, @GenerateinputtxtButton_2Pushed, true);
7822            app.GenerateinputtxtButton_2.HandleVisibility = 'off';
7823            app.GenerateinputtxtButton_2.BackgroundColor = [0.9216 0.9216↙
0.9216];
7824            app.GenerateinputtxtButton_2.FontSize = 15;
7825            app.GenerateinputtxtButton_2.FontWeight = 'bold';
7826            app.GenerateinputtxtButton_2.Layout.Row = 18;
7827            app.GenerateinputtxtButton_2.Layout.Column = [4 10];
7828            app.GenerateinputtxtButton_2.Text = 'Generate input.txt';
7829
7830            % Create UVelocitymeanCheckBox
7831            app.UVelocitymeanCheckBox = uicheckbox(app.GridLayout);
7832            app.UVelocitymeanCheckBox.Tooltip = {'Umean: Mean velocity in X↙
direction'};
7833            app.UVelocitymeanCheckBox.Text = 'U Velocity (mean)';
7834            app.UVelocitymeanCheckBox.Layout.Row = 4;
7835            app.UVelocitymeanCheckBox.Layout.Column = [10 11];
7836
7837            % Create VVelocitymeanCheckBox
7838            app.VVelocitymeanCheckBox = uicheckbox(app.GridLayout);
```

```
7839             app.VVelocitymeanCheckBox.Tooltip = {'Vmean: Mean velocity in Y direction'};
7840             app.VVelocitymeanCheckBox.Text = 'V Velocity (mean)';
7841             app.VVelocitymeanCheckBox.Layout.Row = 5;
7842             app.VVelocitymeanCheckBox.Layout.Column = [10 11];
7843
7844             % Create MinimumWaveHeightCheckBox
7845             app.MinimumWaveHeightCheckBox = uicheckbox(app.GridLayout);
7846             app.MinimumWaveHeightCheckBox.ValueChangedFcn = createCallbackFcn(app, @MinimumWaveHeightCheckBoxValueChanged, true);
7847             app.MinimumWaveHeightCheckBox.Tooltip = {'Hmin: Minimum height within the total simulation time'};
7848             app.MinimumWaveHeightCheckBox.Text = 'Minimum Wave Height';
7849             app.MinimumWaveHeightCheckBox.Layout.Row = 3;
7850             app.MinimumWaveHeightCheckBox.Layout.Column = [2 5];
7851
7852             % Create MaximumWaveHeightCheckBox
7853             app.MaximumWaveHeightCheckBox = uicheckbox(app.GridLayout);
7854             app.MaximumWaveHeightCheckBox.ValueChangedFcn = createCallbackFcn(app, @MaximumWaveHeightCheckBoxValueChanged, true);
7855             app.MaximumWaveHeightCheckBox.Tooltip = {'Hmax: Maximum height within the total simulation time'};
7856             app.MaximumWaveHeightCheckBox.Text = 'Maximum Wave Height';
7857             app.MaximumWaveHeightCheckBox.Layout.Row = 2;
7858             app.MaximumWaveHeightCheckBox.Layout.Column = [2 5];
7859             app.MaximumWaveHeightCheckBox.Value = true;
7860
7861             % Create ThresholdEditFieldLabel
7862             app.ThresholdEditFieldLabel = uilabel(app.GridLayout);
7863             app.ThresholdEditFieldLabel.VerticalAlignment = 'bottom';
7864             app.ThresholdEditFieldLabel.Visible = 'off';
7865             app.ThresholdEditFieldLabel.Layout.Row = 1;
7866             app.ThresholdEditFieldLabel.Layout.Column = 8;
7867             app.ThresholdEditFieldLabel.Text = 'Threshold';
7868
7869             % Create ThresholdEditField
7870             app.ThresholdEditField = uieditfield(app.GridLayout, 'numeric');
7871             app.ThresholdEditField.Limits = [0.0001 Inf];
7872             app.ThresholdEditField.ValueChangedFcn = createCallbackFcn(app, @ThresholdEditFieldValueChanged, true);
7873             app.ThresholdEditField.FontColor = [0.651 0.651 0.651];
7874             app.ThresholdEditField.Visible = 'off';
7875             app.ThresholdEditField.Tooltip = {'Threshold to record the arrival time of the tsunami wave'; ''; 'Unit: meters'};
7876             app.ThresholdEditField.Layout.Row = 2;
7877             app.ThresholdEditField.Layout.Column = 8;
7878             app.ThresholdEditField.Value = 0.0001;
7879
7880             % Create Panel_32
7881             app.Panel_32 = uipanel(app.UIFigure);
7882             app.Panel_32.AutoResizeChildren = 'off';
7883             app.Panel_32.SizeChangedFcn = createCallbackFcn(app, @Panel_32SizeChanged, true);
7884             app.Panel_32.Position = [744 11 398 100];
7885
7886             % Create CreateMakefileButton
7887             app.CreateMakefileButton = uibutton(app.Panel_32, 'push');
7888             app.CreateMakefileButton.ButtonPushedFcn = createCallbackFcn(app,
```

```
@CreateMakefileButtonPushed, true);
7889             app.CreateMakefileButton.VerticalAlignment = 'top';
7890             app.CreateMakefileButton.BackgroundColor = [0.9216 0.9216 0.9216];
7891             app.CreateMakefileButton.FontWeight = 'bold';
7892             app.CreateMakefileButton.Enable = 'off';
7893             app.CreateMakefileButton.Tooltip = {''};
7894             app.CreateMakefileButton.Position = [277 40 106 22];
7895             app.CreateMakefileButton.Text = 'Create Makefile';
7896
7897             % Create StartsimulationButton
7898             app.StartsimulationButton = uibutton(app.Panel_32, 'push');
7899             app.StartsimulationButton.ButtonPushedFcn = createCallbackFcn(app,↙
@StartsimulationButtonPushed, true);
7900             app.StartsimulationButton.BackgroundColor = [0.9216 0.9216↙
0.9216];
7901             app.StartsimulationButton.FontWeight = 'bold';
7902             app.StartsimulationButton.Enable = 'off';
7903             app.StartsimulationButton.Tooltip = {'Compile the Makefile to↙
create ''funwave.exe'' and start the tsunami simulation.'};
7904             app.StartsimulationButton.Position = [277 14 107 22];
7905             app.StartsimulationButton.Text = 'Start simulation';
7906
7907             % Create Button_42
7908             app.Button_42 = uibutton(app.Panel_32, 'push');
7909             app.Button_42.ButtonPushedFcn = createCallbackFcn(app,↙
@Button_42Pushed2, true);
7910             app.Button_42.VerticalAlignment = 'top';
7911             app.Button_42.FontSize = 10;
7912             app.Button_42.FontAngle = 'italic';
7913             app.Button_42.Enable = 'off';
7914             app.Button_42.Tooltip = {''};
7915             app.Button_42.Position = [250 41 19 18];
7916             app.Button_42.Text = '...';
7917
7918             % Create MakefileEditFieldLabel
7919             app.MakefileEditFieldLabel = uilabel(app.Panel_32);
7920             app.MakefileEditFieldLabel.Enable = 'off';
7921             app.MakefileEditFieldLabel.Tooltip = {'Specify the folder↙
directory containing all the .o, .F, and .f90 files'};
7922             app.MakefileEditFieldLabel.Position = [15 39 106 22];
7923             app.MakefileEditFieldLabel.Text = 'Source folder';
7924
7925             % Create MakefileEditField
7926             app.MakefileEditField = uieditfield(app.Panel_32, 'text');
7927             app.MakefileEditField.FontSize = 10;
7928             app.MakefileEditField.FontColor = [0.502 0.502 0.502];
7929             app.MakefileEditField.Enable = 'off';
7930             app.MakefileEditField.Tooltip = {''};
7931             app.MakefileEditField.Placeholder = 'FUNWAVE-TVD-Version_3.6/src';
7932             app.MakefileEditField.Position = [94 42 154 16];
7933
7934             % Create ParallelModeCheckBox
7935             app.ParallelModeCheckBox = uicheckbox(app.Panel_32);
7936             app.ParallelModeCheckBox.Enable = 'off';
7937             app.ParallelModeCheckBox.Tooltip = {'Use parallel mode'; '';↙
'Unchecked box - serial mode'};
7938             app.ParallelModeCheckBox.Text = 'Parallel Mode';
7939             app.ParallelModeCheckBox.Position = [158 66 95 22];
```

```
7940
7941            % Create IntelCompilerCheckBox
7942            app.IntelCompilerCheckBox = uicheckbox(app.Panel_32);
7943            app.IntelCompilerCheckBox.Enable = 'off';
7944            app.IntelCompilerCheckBox.Tooltip = {'Use intel compiler'};
7945            app.IntelCompilerCheckBox.Text = 'Intel Compiler';
7946            app.IntelCompilerCheckBox.Position = [271 66 97 22];
7947
7948            % Create DoublePrecisionCheckBox
7949            app.DoublePrecisionCheckBox = uicheckbox(app.Panel_32);
7950            app.DoublePrecisionCheckBox.ValueChangedFcn = createCallbackFcn↙
(app, @DoublePrecisionCheckBoxValueChanged, true);
7951            app.DoublePrecisionCheckBox.Enable = 'off';
7952            app.DoublePrecisionCheckBox.Tooltip = {'Use double precision'; '';↙
'Unchecked box - single precision'};
7953            app.DoublePrecisionCheckBox.Text = 'Double Precision';
7954            app.DoublePrecisionCheckBox.Position = [26 66 113 22];
7955
7956            % Create Mpif90EditFieldLabel
7957            app.Mpif90EditFieldLabel = uilabel(app.Panel_32);
7958            app.Mpif90EditFieldLabel.Enable = 'off';
7959            app.Mpif90EditFieldLabel.Tooltip = {'Specify the directory↙
containing the MPI F90 Compiler.'; ''; 'Mac OS default:'; '/opt/homebrew/bin/'; '';↙
'Windows OS:'; 'C:\Program Files (x86)\MPICH2\bin'};
7960            app.Mpif90EditFieldLabel.Position = [15 13 77 22];
7961            app.Mpif90EditFieldLabel.Text = 'Mpif90 folder';
7962
7963            % Create Mpif90EditField
7964            app.Mpif90EditField = uieditfield(app.Panel_32, 'text');
7965            app.Mpif90EditField.ValueChangedFcn = createCallbackFcn(app,↙
@Mpif90EditFieldValueChanged, true);
7966            app.Mpif90EditField.ValueChangingFcn = createCallbackFcn(app,↙
@Mpif90EditFieldValueChanging, true);
7967            app.Mpif90EditField.FontSize = 11;
7968            app.Mpif90EditField.FontColor = [0.502 0.502 0.502];
7969            app.Mpif90EditField.Enable = 'off';
7970            app.Mpif90EditField.Tooltip = {''};
7971            app.Mpif90EditField.Placeholder = 'C:\Program Files (x86)↙
\MPICH2\bin\';
7972            app.Mpif90EditField.Position = [94 16 154 16];
7973
7974            % Create Button_41
7975            app.Button_41 = uibutton(app.Panel_32, 'push');
7976            app.Button_41.ButtonPushedFcn = createCallbackFcn(app,↙
@Button_41Pushed, true);
7977            app.Button_41.VerticalAlignment = 'top';
7978            app.Button_41.FontAngle = 'italic';
7979            app.Button_41.Enable = 'off';
7980            app.Button_41.Tooltip = {''};
7981            app.Button_41.Position = [250 15 19 19];
7982            app.Button_41.Text = '...';
7983
7984            % Create Friction_____________________________Label_9
7985            app.Friction_____________________________Label_9 = uilabel(app.↙
UIFigure);
7986            app.Friction_____________________________Label_9.↙
BackgroundColor = [0.9412 0.9412 0.9412];
7987            app.Friction                             Label_9.FontWeight =↙
```

```
'bold';
7988             app.Friction_____________________________Label_9.Tooltip = ↙
{'This section becomes enabled once Makefile-related sections on the left panel are ↙
completed.'};
7989             app.Friction_____________________________Label_9.Position = ↙
[758 100 67 22];
7990             app.Friction_____________________________Label_9.Text = '↙
Makefile  ';
7991 
7992             % Show the figure after all components are created
7993             app.UIFigure.Visible = 'on';
7994         end
7995     end
7996 
7997     % App creation and deletion
7998     methods (Access = public)
7999 
8000         % Construct app
8001         function app = FUNMAP_Input_exported
8002 
8003             runningApp = getRunningApp(app);
8004 
8005             % Check for running singleton app
8006             if isempty(runningApp)
8007 
8008                 % Create UIFigure and components
8009                 createComponents(app)
8010 
8011                 % Register the app with App Designer
8012                 registerApp(app, app.UIFigure)
8013 
8014                 % Execute the startup function
8015                 runStartupFcn(app, @startupFcn)
8016             else
8017 
8018                 % Focus the running singleton app
8019                 figure(runningApp.UIFigure)
8020 
8021                 app = runningApp;
8022             end
8023 
8024             if nargout == 0
8025                 clear app
8026             end
8027         end
8028 
8029         % Code that executes before app deletion
8030         function delete(app)
8031 
8032             % Delete UIFigure when app is deleted
8033             delete(app.UIFigure)
8034         end
8035     end
8036 end
```