```
classdef FUNMAP_Output < matlab.apps.AppBase    2

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                          matlab.ui.Figure
        TabGroup                          matlab.ui.container.TabGroup
        WaveHeightMapTab                  matlab.ui.container.Tab
        GeneralLayoutLabel_3              matlab.ui.control.Label
        Panel_19                          matlab.ui.container.Panel
        ButtonGroup_11                    matlab.ui.container.ButtonGroup
        PlotseparatelyButton_2            matlab.ui.control.RadioButton
        PlotalldatainonefigureButton      matlab.ui.control.RadioButton
        SettoDefaultButton_2              matlab.ui.control.Button
        CloseFiguresButton_4              matlab.ui.control.Button
        FigureSizeLabel_2                 matlab.ui.control.Label
        ThicknessLabel_9                  matlab.ui.control.Label
        Width_2                           matlab.ui.control.NumericEditField
        AutoSetCheckBox_2                 matlab.ui.control.CheckBox
        ThicknessLabel_8                  matlab.ui.control.Label
        Height_2                          matlab.ui.control.NumericEditField
        SouthEditField                    matlab.ui.control.NumericEditField
        SouthEditFieldLabel               matlab.ui.control.Label
        NorthEditField                    matlab.ui.control.NumericEditField
        NorthEditFieldLabel               matlab.ui.control.Label
        WestEditField                     matlab.ui.control.NumericEditField
        WestEditFieldLabel                matlab.ui.control.Label
        EastEditField                     matlab.ui.control.NumericEditField
        EastEditFieldLabel                matlab.ui.control.Label
        PlotindegreesCheckBox_2           matlab.ui.control.CheckBox
        BoundaryLimitsLabel               matlab.ui.control.Label
        GENERATEButton                    matlab.ui.control.Button
        SaveMapLabel                      matlab.ui.control.Label
        Panel_6                           matlab.ui.container.Panel
        mp4CheckBox                       matlab.ui.control.CheckBox
        FramerateEditField_2              matlab.ui.control.NumericEditField
        FramerateEditField_2Label         matlab.ui.control.Label
        OutputDirectoryEditField          matlab.ui.control.EditField
        OutputDirectoryLabel              matlab.ui.control.Label
        Button_2                          matlab.ui.control.Button
        tifCheckBox_2                     matlab.ui.control.CheckBox
        jpgCheckBox                       matlab.ui.control.CheckBox
        FileFormatLabel                   matlab.ui.control.Label
        pngCheckBox                       matlab.ui.control.CheckBox
        BasemapLabel                      matlab.ui.control.Label
        Panel_5                           matlab.ui.container.Panel
        ColorbarTextSize                  matlab.ui.control.NumericEditField
        TextSizeLabel_2                   matlab.ui.control.Label
        CoastlinecolourDropDown_2         matlab.ui.control.DropDown
        FlipBasemapLabel_2                matlab.ui.control.Label
        VerticalCheckBox                  matlab.ui.control.CheckBox
        HorizontalCheckBox                matlab.ui.control.CheckBox
        CoastlinecolourDropDownLabel_9    matlab.ui.control.Label
        mLabel_2                          matlab.ui.control.Label
        mLabel                            matlab.ui.control.Label
        toLabel_2                         matlab.ui.control.Label
        MinEditField_2                    matlab.ui.control.NumericEditField
        MaxEditField_2                    matlab.ui.control.NumericEditField
        DivisionLabel                     matlab.ui.control.Label
```

```
ColorinterpolationEditField_2   matlab.ui.control.NumericEditField
FlipCheckBox_2                  matlab.ui.control.CheckBox
CoastlinecolourDropDown_7       matlab.ui.control.DropDown
etaLabel                        matlab.ui.control.Label
toLabel                         matlab.ui.control.Label
CoastlinecolourDropDownLabel_10  matlab.ui.control.Label
toEditField                     matlab.ui.control.NumericEditField
MinEditField                    matlab.ui.control.NumericEditField
hmaxLabel                       matlab.ui.control.Label
CoastlinecolourDropDownLabel_8  matlab.ui.control.Label
CoastlinecolourDropDown_8       matlab.ui.control.DropDown
FlipCheckBox_5                  matlab.ui.control.CheckBox
DivisionEditField               matlab.ui.control.NumericEditField
DivisionEditFieldLabel          matlab.ui.control.Label
OverlayFeaturesLabel            matlab.ui.control.Label
Panel_7                         matlab.ui.container.Panel
GaugesCheckBox                  matlab.ui.control.CheckBox
ButtonGroup_17                  matlab.ui.container.ButtonGroup
NoneButton                      matlab.ui.control.RadioButton
ArrivalTimeButton               matlab.ui.control.RadioButton
WaveHeightButton                matlab.ui.control.RadioButton
BathymetryButton_2              matlab.ui.control.RadioButton
TabGroup2                       matlab.ui.container.TabGroup
BathymetryTab                   matlab.ui.container.Tab
IntervalEditField_3             matlab.ui.control.NumericEditField
LabelSpacingEditFieldLabel      matlab.ui.control.Label
SpacingEditField_4Label_4       matlab.ui.control.Label
TextLabelSize                   matlab.ui.control.NumericEditField
AddLabelsCheckBox_2             matlab.ui.control.CheckBox
ColorDropDown                   matlab.ui.control.DropDown
ColorDropDownLabel              matlab.ui.control.Label
StyleDropDown                   matlab.ui.control.DropDown
StyleDropDownLabel              matlab.ui.control.Label
LineLabel                       matlab.ui.control.Label
WidthEditField                  matlab.ui.control.NumericEditField
WidthEditFieldLabel             matlab.ui.control.Label
LabelSpacingEditField           matlab.ui.control.NumericEditField
LabelSpacingEditFieldLabel_2    matlab.ui.control.Label
DepthRangeLabel                 matlab.ui.control.Label
IntervalEditField_5             matlab.ui.control.NumericEditField
IntervalEditField_5Label        matlab.ui.control.Label
MinimumEditField                matlab.ui.control.NumericEditField
MinimumEditFieldLabel           matlab.ui.control.Label
MaximumEditField                matlab.ui.control.NumericEditField
MaximumEditFieldLabel           matlab.ui.control.Label
WaveHeightTab                   matlab.ui.container.Tab
wh_interval                     matlab.ui.control.NumericEditField
LabelSpacingEditField_2Label    matlab.ui.control.Label
LabelSpacingEditField_2         matlab.ui.control.NumericEditField
LabelSpacingEditField_2Label_2  matlab.ui.control.Label
SpacingEditField_4Label_5       matlab.ui.control.Label
TextLabelSize_2                 matlab.ui.control.NumericEditField
AddLabelsCheckBox_3             matlab.ui.control.CheckBox
StyleDropDown_2                 matlab.ui.control.DropDown
StyleDropDown_2Label            matlab.ui.control.Label
ColorDropDown_2                 matlab.ui.control.DropDown
ColorDropDown_2Label            matlab.ui.control.Label
ThicknessEditField_2            matlab.ui.control.NumericEditField
```

```
117         ThicknessEditField_2Label       matlab.ui.control.Label
118         LineLabel_2                     matlab.ui.control.Label
119         ContourRangeLabel_2             matlab.ui.control.Label
120         MaximumdepthEditFieldLabel_3    matlab.ui.control.Label
121         MinimumdepthEditFieldLabel_2    matlab.ui.control.Label
122         MaximumdepthEditFieldLabel_2    matlab.ui.control.Label
123         LineintervalEditField_2         matlab.ui.control.NumericEditField
124         MaximumdepthEditField_2         matlab.ui.control.NumericEditField
125         MinimumdepthEditField_2         matlab.ui.control.NumericEditField
126         ArrivalTimeTab                  matlab.ui.container.Tab
127         LabelSpacingEditField_2Label_3  matlab.ui.control.Label
128         LabelSpacingEditField_4         matlab.ui.control.NumericEditField
129         StyleDropDown_4                 matlab.ui.control.DropDown
130         StyleDropDown_4Label_2          matlab.ui.control.Label
131         ColorDropDown_6                 matlab.ui.control.DropDown
132         ColorDropDown_6Label            matlab.ui.control.Label
133         ThicknessEditField_2Label_2     matlab.ui.control.Label
134         ThicknessEditField_3            matlab.ui.control.NumericEditField
135         LineLabel_5                     matlab.ui.control.Label
136         AddLabelsCheckBox_5             matlab.ui.control.CheckBox
137         SpacingEditField_4Label_6       matlab.ui.control.Label
138         TextLabelSize_3                 matlab.ui.control.NumericEditField
139         IntervalEditField_7             matlab.ui.control.NumericEditField
140         LabelSpacingEditField_2Label_4  matlab.ui.control.Label
141         MinimumdepthEditFieldLabel_5    matlab.ui.control.Label
142         MinimumdepthEditField_3         matlab.ui.control.NumericEditField
143         MaximumdepthEditFieldLabel_10   matlab.ui.control.Label
144         LineintervalEditField_3         matlab.ui.control.NumericEditField
145         MaximumdepthEditFieldLabel_9    matlab.ui.control.Label
146         MaximumdepthEditField_3         matlab.ui.control.NumericEditField
147         ContourRangeLabel               matlab.ui.control.Label
148         FileEditField_3                 matlab.ui.control.EditField
149         FileEditField_3Label            matlab.ui.control.Label
150         TimeFileButton                  matlab.ui.control.Button
151         GaugesTab                       matlab.ui.container.Tab
152         GLspacing                       matlab.ui.control.NumericEditField
153         GLspacingLabel                  matlab.ui.control.Label
154         SizeEditField_3                 matlab.ui.control.NumericEditField
155         SizeEditField_3Label            matlab.ui.control.Label
156         CoastlinecolourDropDown_5       matlab.ui.control.DropDown
157         CoastlinecolourDropDown_4       matlab.ui.control.DropDown
158         AlignmentLabel                  matlab.ui.control.Label
159         AddLabelsCheckBox_4             matlab.ui.control.CheckBox
160         SizeEditField                   matlab.ui.control.NumericEditField
161         SizeEditFieldLabel              matlab.ui.control.Label
162         CoastlinecolourDropDown_3       matlab.ui.control.DropDown
163         CoastlinecolourDropDownLabel_3  matlab.ui.control.Label
164         ColorDropDown_3                 matlab.ui.control.DropDown
165         ColorDropDown_3Label            matlab.ui.control.Label
166         FileEditField                   matlab.ui.control.EditField
167         GaugeFileLabel                  matlab.ui.control.Label
168         Button_8                        matlab.ui.control.Button
169         Tab                             matlab.ui.container.Tab
170         InputDataLabel                  matlab.ui.control.Label
171         Panel_3                         matlab.ui.container.Panel
172         gridY                           matlab.ui.control.NumericEditField
173         YEditField_2Label               matlab.ui.control.Label
174         gridX                           matlab.ui.control.NumericEditField
```

```
175         XEditField_2Label                matlab.ui.control.Label
176         GridSizeLabel_2                  matlab.ui.control.Label
177         StartLabel                       matlab.ui.control.Label
178         StartTime                        matlab.ui.control.NumericEditField
179         IntervalLabel                    matlab.ui.control.Label
180         TotalSimuilationTimesecEditField_3  matlab.ui.control.NumericEditField
181         TotalSimuilationTimesecEditFieldLabel_3  matlab.ui.control.Label
182         BathymetryLabel                  matlab.ui.control.Label
183         FilesLabel                       matlab.ui.control.Label
184         DepthFileEditField               matlab.ui.control.EditField
185         Button_15                        matlab.ui.control.Button
186         LatitudeEditField                matlab.ui.control.NumericEditField
187         LatitudeEditFieldLabel           matlab.ui.control.Label
188         LongitudeEditField               matlab.ui.control.NumericEditField
189         LongitudeEditFieldLabel          matlab.ui.control.Label
190         SouthwestCornerLabel             matlab.ui.control.Label
191         Button_14                        matlab.ui.control.Button
192         FileTextArea                     matlab.ui.control.TextArea
193         GaugeRecordsTab                  matlab.ui.container.Tab
194         PlotStyleandLayoutLabel          matlab.ui.control.Label
195         SavePlotLabel                    matlab.ui.control.Label
196         Panel_13                         matlab.ui.container.Panel
197         epsCheckBox                      matlab.ui.control.CheckBox
198         txtCheckBox                      matlab.ui.control.CheckBox
199         OutputDirectoryEditField_2       matlab.ui.control.EditField
200         OutputDirectoryLabel_3           matlab.ui.control.Label
201         Button_18                        matlab.ui.control.Button
202         pdfCheckBox_2                    matlab.ui.control.CheckBox
203         jpgCheckBox_2                    matlab.ui.control.CheckBox
204         FileFormatLabel_2                matlab.ui.control.Label
205         pngCheckBox_2                    matlab.ui.control.CheckBox
206         InputDataLabel_3                 matlab.ui.control.Label
207         Panel_11                         matlab.ui.container.Panel
208         XaxisUseDataDropDown             matlab.ui.control.DropDown
209         YaxisUseDataDropDown             matlab.ui.control.DropDown
210         YaxisSetcolumncountLabel         matlab.ui.control.Label
211         Label_11                         matlab.ui.control.Label
212         Label_10                         matlab.ui.control.Label
213         Label_7                          matlab.ui.control.Label
214         Label_6                          matlab.ui.control.Label
215         Label_5                          matlab.ui.control.Label
216         denomeratorX                     matlab.ui.control.NumericEditField
217         numeratorX                       matlab.ui.control.NumericEditField
218         CheckBox_4                       matlab.ui.control.CheckBox
219         CheckBox_3                       matlab.ui.control.CheckBox
220         YaxisEditFieldLabel              matlab.ui.control.Label
221         Label_9                          matlab.ui.control.Label
222         denomeratorY                     matlab.ui.control.NumericEditField
223         numeratorY                       matlab.ui.control.NumericEditField
224         ConvertLabel                     matlab.ui.control.Label
225         XaxisdataLabel                   matlab.ui.control.Label
226         SelectColumnLabel                matlab.ui.control.Label
227         XaxisLabel                       matlab.ui.control.Label
228         Button_16                        matlab.ui.control.Button
229         FileTextArea_2                   matlab.ui.control.TextArea
230         FilesLabel_2                     matlab.ui.control.Label
231         PLOTButton                       matlab.ui.control.Button
232         Panel_10                         matlab.ui.container.Panel
```

```
233         MarkerDropDown                matlab.ui.control.DropDown
234         MarkerDropDownLabel           matlab.ui.control.Label
235         ThicknessLabel_4              matlab.ui.control.Label
236         LegendSize                    matlab.ui.control.NumericEditField
237         LocationDropDown              matlab.ui.control.DropDown
238         LocationDropDownLabel         matlab.ui.control.Label
239         legendfirsttext               matlab.ui.control.EditField
240         legendfirsttextlabel          matlab.ui.control.Label
241         LegendLabel                   matlab.ui.control.Label
242         ThicknessLabel_12             matlab.ui.control.Label
243         LineYThickness_2              matlab.ui.control.NumericEditField
244         CloseFiguresButton_2          matlab.ui.control.Button
245         AutoSetCheckBox               matlab.ui.control.CheckBox
246         ThicknessLabel_7              matlab.ui.control.Label
247         Width                         matlab.ui.control.NumericEditField
248         FigureSizeLabel               matlab.ui.control.Label
249         ThicknessLabel_6              matlab.ui.control.Label
250         Height                        matlab.ui.control.NumericEditField
251         FlipCheckBox_3                matlab.ui.control.CheckBox
252         GaugeLineColor_multiple       matlab.ui.control.DropDown
253         GridLabelSize                 matlab.ui.control.NumericEditField
254         ThicknessLabel_5              matlab.ui.control.Label
255         ThicknessLabel_3              matlab.ui.control.Label
256         LineYThickness                matlab.ui.control.NumericEditField
257         Lineaty0DropDown              matlab.ui.control.DropDown
258         Lineaty0DropDownLabel         matlab.ui.control.Label
259         GridStyle                     matlab.ui.control.DropDown
260         StyleDropDown_4Label          matlab.ui.control.Label
261         BackgroundGridLabel           matlab.ui.control.Label
262         XAxisEditField                matlab.ui.control.EditField
263         YAxisLabelEditField_2Label    matlab.ui.control.Label
264         XandYLimitsLabel_2            matlab.ui.control.Label
265         YAxisEditField                matlab.ui.control.EditField
266         LabelEditFieldLabel           matlab.ui.control.Label
267         XAutoSetLabel                 matlab.ui.control.Label
268         YAutoSetLabel                 matlab.ui.control.Label
269         CheckBox                      matlab.ui.control.CheckBox
270         yMaxLimit                     matlab.ui.control.NumericEditField
271         MinimumdepthEditFieldLabel_3  matlab.ui.control.Label
272         yMinLimit                     matlab.ui.control.NumericEditField
273         MaximumdepthEditFieldLabel_8  matlab.ui.control.Label
274         yLimInterval                  matlab.ui.control.NumericEditField
275         MaximumdepthEditFieldLabel_4  matlab.ui.control.Label
276         CheckBox_2                    matlab.ui.control.CheckBox
277         MaximumdepthEditFieldLabel_6  matlab.ui.control.Label
278         xLimInterval                  matlab.ui.control.NumericEditField
279         MaximumdepthEditFieldLabel_5  matlab.ui.control.Label
280         xMaxLimit                     matlab.ui.control.NumericEditField
281         MinimumdepthEditFieldLabel_4  matlab.ui.control.Label
282         xMinLimit                     matlab.ui.control.NumericEditField
283         TextArea2_4                   matlab.ui.control.TextArea
284         AxesLabels                    matlab.ui.control.Label
285         WidthLabel                    matlab.ui.control.Label
286         DataLineWidth                 matlab.ui.control.NumericEditField
287         ButtonGroup_10                matlab.ui.container.ButtonGroup
288         PlotseparatelyButton_4        matlab.ui.control.RadioButton
289         PlotalldatainonegraphButton   matlab.ui.control.RadioButton
290         GaugeLineStyle                matlab.ui.control.DropDown
```

```
291         LineStyleLabel                  matlab.ui.control.Label
292         GaugeLineColor                  matlab.ui.control.DropDown
293         ColorDropDownLabel_2            matlab.ui.control.Label
294         PlotPropertiesLabel             matlab.ui.control.Label
295         VelocityMapTab                  matlab.ui.container.Tab
296         GeneralLayoutLabel_2            matlab.ui.control.Label
297         Panel_18                        matlab.ui.container.Panel
298         ButtonGroup_12                  matlab.ui.container.ButtonGroup
299         PlotseparatelyButton_3          matlab.ui.control.RadioButton
300         PlotalldatainonefigureButton_2  matlab.ui.control.RadioButton
301         SettoDefaultButton              matlab.ui.control.Button
302         CloseFiguresButton_3            matlab.ui.control.Button
303         FigureSizeLabel_3               matlab.ui.control.Label
304         ThicknessLabel_11               matlab.ui.control.Label
305         Width_3                         matlab.ui.control.NumericEditField
306         AutoSetCheckBox_3               matlab.ui.control.CheckBox
307         ThicknessLabel_10               matlab.ui.control.Label
308         Height_3                        matlab.ui.control.NumericEditField
309         SouthEditField_2                matlab.ui.control.NumericEditField
310         SouthEditField_2Label           matlab.ui.control.Label
311         NorthEditField_2                matlab.ui.control.NumericEditField
312         NorthEditField_2Label           matlab.ui.control.Label
313         WestEditField_2                 matlab.ui.control.NumericEditField
314         WestEditField_2Label            matlab.ui.control.Label
315         EastEditField_2                 matlab.ui.control.NumericEditField
316         EastEditField_2Label            matlab.ui.control.Label
317         PlotindegreesCheckBox           matlab.ui.control.CheckBox
318         BoundaryLimitsLabel_2           matlab.ui.control.Label
319         BasemapandOverlaysLabel         matlab.ui.control.Label
320         SaveMapLabel_2                  matlab.ui.control.Label
321         Panel_17                        matlab.ui.container.Panel
322         tifCheckBox                     matlab.ui.control.CheckBox
323         txtCheckBox_2                   matlab.ui.control.CheckBox
324         OutputDirectoryEditField_3      matlab.ui.control.EditField
325         OutputDirectoryLabel_2          matlab.ui.control.Label
326         mp4CheckBox_2                   matlab.ui.control.CheckBox
327         FramerateEditField              matlab.ui.control.NumericEditField
328         FramerateEditFieldLabel         matlab.ui.control.Label
329         Button_26                       matlab.ui.control.Button
330         jpgCheckBox_3                   matlab.ui.control.CheckBox
331         FileFormatLabel_3               matlab.ui.control.Label
332         pngCheckBox_3                   matlab.ui.control.CheckBox
333         Panel_16                        matlab.ui.container.Panel
334         TabGroup3                       matlab.ui.container.TabGroup
335         BasemapTab                      matlab.ui.container.Tab
336         LandColor                       matlab.ui.control.DropDown
337         CoastlinecolourDropDownLabel_12  matlab.ui.control.Label
338         FlipBasemapLabel                matlab.ui.control.Label
339         VerticalCheckBox_2              matlab.ui.control.CheckBox
340         HorizontalCheckBox_2            matlab.ui.control.CheckBox
341         FileTextArea_5                  matlab.ui.control.TextArea
342         ColorbarLabel                   matlab.ui.control.Label
343         ColormapLabel                   matlab.ui.control.Label
344         ThirdTabColorMapPanel           matlab.ui.container.Panel
345         TextSizeLabel                   matlab.ui.control.Label
346         LimitsLabel                     matlab.ui.control.Label
347         MaxBarValue                     matlab.ui.control.NumericEditField
348         mLabel_3                        matlab.ui.control.Label
```

```
349         toLabel_3                         matlab.ui.control.Label
350         MinBarValue                       matlab.ui.control.NumericEditField
351         InterpolationDivisionEditField_2  matlab.ui.control.NumericEditField
352         InterpolationDivisionEditFieldLabel_2  matlab.ui.control.Label
353         FlipCheckBox_4                    matlab.ui.control.CheckBox
354         BackgroundMapColorDropDown        matlab.ui.control.DropDown
355         maplabelsize                      matlab.ui.control.NumericEditField
356         ButtonGroup_16                    matlab.ui.container.ButtonGroup
357         VorticityButton                   matlab.ui.control.RadioButton
358         BathymetryButton                  matlab.ui.control.RadioButton
359         hmaxButton                        matlab.ui.control.RadioButton
360         etaButton                         matlab.ui.control.RadioButton
361         VelocityButton                    matlab.ui.control.RadioButton
362         BathymetryContoursTab_2           matlab.ui.container.Tab
363         IntervalEditField_8               matlab.ui.control.NumericEditField
364         SpacingEditField                  matlab.ui.control.NumericEditField
365         SpacingEditField_4Label_2         matlab.ui.control.Label
366         SpacingEditField_4Label_3         matlab.ui.control.Label
367         LabelSizeCont                     matlab.ui.control.NumericEditField
368         IntervalEditField_8Label          matlab.ui.control.Label
369         WidthEditField_2                  matlab.ui.control.NumericEditField
370         WidthEditField_2Label             matlab.ui.control.Label
371         StyleDropDown_3                   matlab.ui.control.DropDown
372         StyleDropDown_3Label_2            matlab.ui.control.Label
373         ColorDropDown_4                   matlab.ui.control.DropDown
374         ColorDropDown_4Label              matlab.ui.control.Label
375         AddLabelCheckBox_2                matlab.ui.control.CheckBox
376         LineLabel_4                       matlab.ui.control.Label
377         IntervalEditField_6               matlab.ui.control.NumericEditField
378         IntervalEditField_6Label          matlab.ui.control.Label
379         MaximumEditField_2                matlab.ui.control.NumericEditField
380         MaximumEditField_2Label           matlab.ui.control.Label
381         MinimumEditField_2                matlab.ui.control.NumericEditField
382         MinimumEditField_2Label           matlab.ui.control.Label
383         PlotBathymetryContoursCheckBox    matlab.ui.control.CheckBox
384         RangeLabel                        matlab.ui.control.Label
385         GaugesTab_2                       matlab.ui.container.Tab
386         SizeEditField_2                   matlab.ui.control.NumericEditField
387         SizeEditField_2Label              matlab.ui.control.Label
388         CoastlinecolourDropDown_11        matlab.ui.control.DropDown
389         CoastlinecolourDropDown_10        matlab.ui.control.DropDown
390         AlignmentLabel_2                  matlab.ui.control.Label
391         GLspacingLabel_2                  matlab.ui.control.Label
392         GLspacing_2                       matlab.ui.control.NumericEditField
393         FontSizeEditField_2               matlab.ui.control.NumericEditField
394         FontSizeEditField_2Label          matlab.ui.control.Label
395         AddLabelCheckBox                  matlab.ui.control.CheckBox
396         PlotGaugesCheckBox                matlab.ui.control.CheckBox
397         FileEditField_2                   matlab.ui.control.EditField
398         GaugeFileLabel_2                  matlab.ui.control.Label
399         Button_27                         matlab.ui.control.Button
400         ColorDropDown_7                   matlab.ui.control.DropDown
401         ColorDropDown_7Label              matlab.ui.control.Label
402         gaugemarkerVelocityTab            matlab.ui.control.DropDown
403         CoastlinecolourDropDownLabel_19   matlab.ui.control.Label
404         ArrowsLabel                       matlab.ui.control.Label
405         InputDataLabel_4                  matlab.ui.control.Label
406         Panel_15                          matlab.ui.container.Panel
```

```
        Button_19                          matlab.ui.control.Button
        FileTextArea_4                     matlab.ui.control.TextArea
        gridX_2                            matlab.ui.control.NumericEditField
        XEditField_2Label_2                matlab.ui.control.Label
        TotalSimuilationTimesecEditField_4  matlab.ui.control.NumericEditField
        IntervalLabel_2                    matlab.ui.control.Label
        StartLabel_2                       matlab.ui.control.Label
        StartTime2                         matlab.ui.control.NumericEditField
        TotalSimuilationTimesecEditFieldLabel_4  matlab.ui.control.Label
        gridY_2                            matlab.ui.control.NumericEditField
        YEditField_2Label_2                matlab.ui.control.Label
        GridSizeLabel                      matlab.ui.control.Label
        LatitudeEditField_2                matlab.ui.control.NumericEditField
        LatitudeEditField_2Label           matlab.ui.control.Label
        LongitudeEditField_2               matlab.ui.control.NumericEditField
        LongitudeEditField_2Label          matlab.ui.control.Label
        SouthwestCornerLabel_2             matlab.ui.control.Label
        Button_22                          matlab.ui.control.Button
        BathymetryLabel_2                  matlab.ui.control.Label
        DepthFileEditField_2               matlab.ui.control.EditField
        UvectorsLabel                      matlab.ui.control.Label
        FileTextArea_3                     matlab.ui.control.TextArea
        Panel_14                           matlab.ui.container.Panel
        ArrowSpacing                       matlab.ui.control.NumericEditField
        SpacingLabel                       matlab.ui.control.Label
        PlotvectorsCheckBox                matlab.ui.control.CheckBox
        ArrowHeadSize                      matlab.ui.control.NumericEditField
        ArrowSizeLabel                     matlab.ui.control.Label
        ColorLabel_2                       matlab.ui.control.Label
        ArrowThickness                     matlab.ui.control.NumericEditField
        ArrowThicknessLabel                matlab.ui.control.Label
        arrowscale                         matlab.ui.control.NumericEditField
        XEditField_2Label_3                matlab.ui.control.Label
        QuiverColorDropDown                matlab.ui.control.DropDown
        GENERATEButton_2                   matlab.ui.control.Button
    end


    properties (Access = private)
        ArrivalTimeConverted
        ArrivalTimename
        bathyCol
        bathymetrycheck
        bathyRow
        colFirstFile
        ETAfilenames
        ETAfullfile
        FigureDirectoryVector
        FileINPUT
        FileINPUT1
        FileINPUT2
        FileINPUT3
        FileINPUT4
        FileNAME
        FileNAME1
        FileNAME2
        FileNAME3
        FileNAME4
```

```
465         FileNAME5
466         GAUGEFILE
467         GAUGEFILE2
468         GAUGEFILE3
469         GaugeDirectorylist
470         GaugeFname
471         GaugeFname3
472         GaugeNames
473         HMAXfilenames
474         HMAXfullfile
475         INfile
476         LegendGauge
477         LegendLocation
478         MainDirectoryVector
479         STATION_FILE
480         Ufiles
481         Uvectorfullfile
482         Vfile
483         Vvectorfilenames
484         Vvectorfullfile
485         WORKFOLDER0
486         WORKFOLDER
487         WORKFOLDER2
488         WORKFOLDER3
489         WORKFOLDER4
490         WORKFOLDER5
491         XCOLGAUGE
492         YCOLGAUGE
493         bathymetrydata
494         bathymetrydata2
495         bathymetryinputdata
496         bathymetryinputdata2
497         bathymetryname
498         bathymetryname2
499         col1
500         col1vel
501         colorcombi
502         fin
503         inputtxt_directory0
504         inputtxt_directory1
505         LandAreaColor
506         latGauge
507         latGauge2
508         longGauge
509         longGauge2
510         row1
511         row1vel
512         rowFirstFile
513         vectorpath
514         x
515         xcoord
516         y
517         ycoord
518     end
519
520
521     % Callbacks that handle component events
522     methods (Access = private)
```

```
523
524         % Code that executes after component creation
525         function startupFcn(app)
526      %Set the screenshot mode to 'manual' to disable automatic screenshots,
527      %to improve app efficiency
528             currentAppModel.MetadataModel.ScreenshotMode = 'manual';
529
530         end
531
532         % Button pushed function: Button_14
533         function Button_14Pushed(app, event)
534            %Disable all the buttons
535             app.CoastlinecolourDropDown_8.Enable = "off";
536             app.FlipCheckBox_5.Enable = "off";
537             app.toEditField.Enable = "off";
538             app.MinEditField.Enable = "off";
539             app.CoastlinecolourDropDown_7.Enable = "off";
540             app.FlipCheckBox_2.Enable = "off";
541             app.MaxEditField_2.Enable = "off";
542             app.MinEditField_2.Enable = "off";
543             app.DivisionEditField.Enable = "off";
544             app.DivisionEditFieldLabel.Enable = "off";
545             app.DivisionLabel.Enable = "off";
546             app.ColorinterpolationEditField_2.Enable = "off";
547             drawnow
548
549
550             %Find files to import from a directory
551             [files, path] = uigetfile('*.*', 'Select the files',↙
'MultiSelect', 'on');
552             if isequal(files, 0) || isequal(path, 0)
553                 return;
554             end
555
556             cd(fullfile(path))
557
558             if ischar(files) % Only 1 file is uploaded
559                 files = {files};
560             end
561
562             %Sort the files by snapshot timing, and display the list of↙
filenames in the app's textbox
563             app.FileNAME = sort(string(files));
564             allfiles = fullfile(path, app.FileNAME);
565             app.FileINPUT = allfiles;
566             app.FileTextArea.Value = app.FileNAME; %
567             app.FileTextArea.FontColor = 'k';
568
569
570             %Set variables for tracking the file types
571             containsHmax = false;
572             containsEta = false;
573             containsTime = false;
574
575             %Verify that each file is readable
576             for i = 1:length(allfiles)
577                 if exist(allfiles{i}, 'file') ~= 2
578                     errordlg(['Unable to find or open file: ' allfiles{i}],↙
```

```
'File Error', 'modal');
                    continue; % Skip to the next file
                end

                %Update T/F statement for file-type tracking
                str = app.FileNAME(i);
                if contains(str, "hmax")
                    containsHmax = true;
                end
                if contains(str, "eta")
                    containsEta = true;
                end
                if contains(str, "time")
                    containsTime = true;
                end
            end

            %Enable or disable colormap options based on file type
            if containsHmax || containsTime
                app.CoastlinecolourDropDown_8.Enable = "on";
                app.FlipCheckBox_5.Enable = "on";
                app.toEditField.Enable = "on";
                app.MinEditField.Enable = "on";
                app.ColorbarTextSize.Enable = "on";
                app.DivisionEditFieldLabel.Enable = "on";
                app.DivisionEditField.Enable = "on";
            end
            if containsEta
                app.CoastlinecolourDropDown_7.Enable = "on";
                app.FlipCheckBox_2.Enable = "on";
                app.MaxEditField_2.Enable = "on";
                app.MinEditField_2.Enable = "on";
                app.ColorinterpolationEditField_2.Enable = "on";
                app.DivisionLabel.Enable = "on";
                app.ColorbarTextSize.Enable = "on";
            else
                app.CoastlinecolourDropDown_8.Enable = "on";
                app.FlipCheckBox_5.Enable = "on";
                app.toEditField.Enable = "on";
                app.MinEditField.Enable = "on";
                app.ColorbarTextSize.Enable = "on";
                app.DivisionEditFieldLabel.Enable = "on";
                app.DivisionEditField.Enable = "on";

            end


            %Read the first file to extract its matrix size, which will be↙
used to set up the map boundary limit
            try
                f = readmatrix(app.FileNAME(1));
                [row,col] = size(f);
                app.col1 = col-1;
                app.row1 = row-1;
            end
        end

        % Button pushed function: GENERATEButton
```

```
635         function GENERATEButtonPushed(app, event)
636 %Please cite this paper when you use the applications:
637 %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
638
639
640
641
642 % NANYANG TECHNOLOGICAL UNIVERSITY - NTUITIVE PTE LTD (NTUITIVE) Dual License Agreement
643 % Non-Commercial Use Only
644 % 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").
645 %
646 % 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.
647 %
648 % SCOPE OF RIGHTS:
649 % 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.
650 % 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.
651 % 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.
652 % 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.
653 % 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.
654 %
655 % You may not distribute this Software or any derivative works.
656 % In return, we simply require that you agree:
657 % 1.    That you will not remove any copyright or other notices from the Software.
```

```
658 % 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.
659 % 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.
660 % 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.
661 % 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.
662 % 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.
663 % 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.
664 % 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.
665 % 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.
666 % 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.
667 % 11.   That all rights not expressly granted to you in this NTUITIVE-LA are
reserved.
668 % 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.
669 %
670 %
671 % 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.”
672 %
673 % Copyright (c) NTUITIVE. All rights reserved.
674
675
676
677 %A copy of FUNWAVE's BDS license agreement:
678 % !    Copyright (c) 2016, FUNWAVE Development Team
```

```
 679 % !
 680 % !    (See http://www.udel.edu/kirby/programs/funwave/funwave.html
 681 % !     for Development Team membership)
 682 % !
 683 % !    All rights reserved.
 684 % !
 685 % !    FUNWAVE_TVD is free software: you can redistribute it and/or modify
 686 % !    it under the terms of the Simplified BSD License as released by
 687 % !    the Berkeley Software Distribution (BSD).
 688 % !
 689 % !    Redistribution and use in source and binary forms, with or without
 690 % !    modification, are permitted provided that the following conditions are↙
met:
 691 % !
 692 % !    1. Redistributions of source code must retain the above copyright↙
notice, this
 693 % !       list of conditions and the following disclaimer.
 694 % !    2. Redistributions in binary form must reproduce the above copyright↙
notice,
 695 % !    this list of conditions and the following disclaimer in the↙
documentation
 696 % !    and/or other materials provided with the distribution.
 697 % !
 698 % !    THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS↙
IS" AND
 699 % !    ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE↙
IMPLIED
 700 % !    WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 701 % !    DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE↙
LIABLE FOR
 702 % !    ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL↙
DAMAGES
 703 % !    (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR↙
SERVICES;
 704 % !    LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED↙
AND
 705 % !    ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR↙
TORT
 706 % !    (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE↙
OF THIS
 707 % !    SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 708 % !
 709 % !    The views and conclusions contained in the software and documentation↙
are those
 710 % !    of the authors and should not be interpreted as representing official↙
policies,
 711 % !    either expressed or implied, of the FreeBSD Project.
 712
 713
 714
 715 %This is the start of the app's code:
 716 close all
 717
 718             %--------------------------
 719             %------BASEMAP INPUT------
 720             %--------------------------
 721             %List of the  colormap items in the dropdown list
 722             keys = {'autumn', 'bone', 'colorcube', 'cool', 'copper', 'flag',↙
```

```
'gray', 'hot', 'hsv', 'jet', ...
 723                 'parula', 'pink', 'prism', 'spring', 'summer', 'turbo',↙
'winter', 'blue', 'blue - green', ...
 724                 'blue - purple', 'green - blue', 'greens', 'grays', 'oranges',↙
'orange - red', 'purple - blue', ...
 725                 'purple - blue -green', 'purple - red', 'purples', 'red -↙
purple', 'reds', 'yellow - green', ...
 726                 'yellow - green - blue', 'yellow - orange - brown', 'yellow -↙
orange - red', 'brown - teal', ...
 727                 'pink - light green', 'purple - green', 'purple - orange',↙
'red - blue', 'red - gray', ...
 728                 'red - yellow - blue', 'red - yellow - green', 'spectral',↙
'accent', 'dark 2', 'paired', ...
 729                 'pastel 1', 'pastel 2', 'set 1', 'set 2', 'set 3', '--- MATLAB↙
default ----', '---  CBREWER 2 ---', ...
 730                 '< sequential >', '< divergent >', '< qualitative >'};
 731
 732             %The equivalent colormap code
 733             keysColor = {'autumn', 'bone', 'colorcube', 'cool', 'copper',↙
'flag', 'gray', 'hot', 'hsv', 'jet', ...
 734                 'parula', 'pink', 'prism', 'spring', 'summer', 'turbo',↙
'winter','Blues','BuGn','BuPu','GnBu','Greens', ...
 735                 'Greys','Oranges','OrRd','PuBu','PuBuGn', ...
 736↙
'PuRd','Purples','RdPu','Reds','YlGn','YlGnBu','YlOrBr','YlOrRd','BrBG','PiYG','PRG↙
n','PuOr','RdBu','RdGy', ...
 737                 'RdYlBu','RdYlGn','Spectral','Accent', ...
 738↙
'Dark2','Paired','Pastel1','Pastel2','Set1','Set2','Set3','parula','parula','parula↙
','parula','parula'};
 739
 740             %The colorbrewer
 741             keysBrewer = {'none', 'none', 'none', 'none', 'none', 'none',↙
'none', 'none', 'none', 'none', ...
 742                 'none', 'none', 'none', 'none', 'none', 'none', 'none', 'seq',↙
'seq', ...
 743                 'seq', 'seq', 'seq', 'seq', 'seq', 'seq', ...
 744                 'seq', 'seq', 'seq', 'seq', 'seq', ...
 745                 'seq', 'seq', 'seq', 'seq', 'seq', 'div', 'div', 'div', ...
 746                 'div', 'div', 'div', 'div', 'div', ...
 747                 'div', 'qual', 'qual', 'qual', ...
 748                 'qual', 'qual', 'qual', 'qual', 'qual', ...
 749                 'none', 'none', 'none','none', 'none'};
 750
 751             % Match dropdown list items to their corresponding colors and↙
ColorBrewer
 752             colorMap = containers.Map(keys, keysColor);
 753             colorBrewer = containers.Map(keys, keysBrewer);
 754
 755             %For eta colormap
 756             datacolorETA = colorMap(app.CoastlinecolourDropDown_7.Value);
 757             datacolorbrewerETA = colorBrewer(app.CoastlinecolourDropDown_7.↙
Value);
 758
 759             %For hmax colormap
 760             datacolor = colorMap(app.CoastlinecolourDropDown_8.Value);
 761             datacolorbrewer  = colorBrewer(app.CoastlinecolourDropDown_8.↙
Value);
```

```
762
763
764             %--------------------------
765             %------LAND AREA COLOR------
766             %--------------------------
767
768             %List of commonly used colors and styles
769             colorMap = containers.Map({'Black', 'Dark gray', 'Medium gray',↙
'Light gray', 'Red', 'Green', 'Blue', 'Yellow', 'Cyan', 'Magenta', 'White'}, {'k',↙
'[0.4 0.4 0.4]', '[0.3 0.3 0.3]', '[0.8 0.8 0.8]', 'r', 'g', 'b', 'y', 'c', 'm',↙
'w'}); %[0.7 0.7 0.7]
770             lineStyleMap = containers.Map({'Solid', 'Dashed', 'Dotted', 'Dash-↙
dotted'}, {'-', '--', ':', '-.'});
771
772             %Landmass color
773             app.LandAreaColor = colorMap(app.CoastlinecolourDropDown_2.Value);
774
775             %------------------
776             %------GAUGES------
777             %------------------
778             %Marker Style
779             GaugeMarkerOptions = containers.Map({'none', 'o', '+', '*', '.',↙
'x', '-', '|', '^', 'v', '>', '<', 'diamond', 'hexagram', 'pentagram', 'square'},↙
...
780                 {'none', 'o', '+', '*', '.', 'x', '-', '|', '^', 'v', '>',↙
'<', 'diamond', 'hexagram', 'pentagram', 'square'});
781             GaugeMarker = GaugeMarkerOptions(app.CoastlinecolourDropDown_3.↙
Value);
782
783             %Marker Color
784             GaugeColor =  colorMap(app.ColorDropDown_3.Value);
785
786             %Label - horizontal alignment
787             horzLabelMap = containers.Map({'Centre', 'Right', 'Left'},↙
{'center', 'left', 'right'});
788             HorzLabel = horzLabelMap(app.CoastlinecolourDropDown_4.Value);
789
790             %Label - vertical alignment
791             vertLabelMap = containers.Map({'Centre', 'Top', 'Bottom'},↙
{'middle', 'bottom', 'top'});
792             VertLabel = vertLabelMap(app.CoastlinecolourDropDown_5.Value);
793
794
795             %--------------------------------
796             %------BATHYMETRY CONTOURS------
797             %--------------------------------
798             %Line Colours
799             ContourColor = colorMap(app.ColorDropDown.Value);
800             %Line style
801             ContourLineStyle = lineStyleMap(app.StyleDropDown.Value);
802
803
804             %--------------------------------
805             %------WAVE HEIGHT CONTOURS------
806             %--------------------------------
807             %Line Colours
808             WaveContourColor = colorMap(app.ColorDropDown_2.Value);
809             %Line style
```

```
            WaveContourLineStyle = lineStyleMap(app.StyleDropDown_2.Value);


            %---------------------------------------
            %------FIRST WAVE ARRIVAL CONTOURS------
            %---------------------------------------
            %Default threshold: 0.001 m
            %Line Colours
            ArrivalTimeContourColor = colorMap(app.ColorDropDown_6.Value);
            %Line Style
            ArrivalTimeLineStyle = lineStyleMap(app.StyleDropDown_4.Value);

            %---------------------------------------
            %------SAVE MAP: Output Directory------
            %---------------------------------------
            %Check if the output directory has been manually set by the user
            if app.OutputDirectoryEditField.Value == string(app.WORKFOLDER0)
                app.inputtxt_directory0  = fullfile(app.WORKFOLDER);
                app.inputtxt_directory1 = fullfile(app.WORKFOLDER0);

            else % Set the default directory to the Desktop
                if ismac  %macOS
                    defaultDir = fullfile(getenv('HOME'), 'Desktop');
                elseif ispc  %Windows
                    defaultDir = fullfile(getenv('USERPROFILE'), 'Desktop');
                else %Others
                    defaultDir = pwd;
                end

                %Create 'OUTPUT_FILES' folder and 'Figures' subfolder
                Dir0 = fullfile(defaultDir, 'OUTPUT_FILES');
                Dir1 = fullfile(defaultDir, 'OUTPUT_FILES/Figures');
                if ~exist(Dir0, 'dir')
                    mkdir(Dir0);
                end
                if ~exist(Dir1, 'dir')
                    mkdir(Dir1);
                end

                app.inputtxt_directory0 = Dir0;
                app.inputtxt_directory1 = Dir1;

                %Display the path directory of the 'Figures' subfolder in the↙
textbox
                app.OutputDirectoryEditField.Value = deblank(string(Dir1));
                app.OutputDirectoryEditField.FontColor = 'k';
            end


            %Create animation/ creating and opening the file
            if app.mp4CheckBox.Value
                vidObj = VideoWriter(fullfile(char(app.inputtxt_directory1),↙
'animation_waveheight.mp4'), 'MPEG-4');
                vidObj.FrameRate = app.FramerateEditField_2.Value;
                open(vidObj);
            end

```

```
866             %----------------------
867             %------LOG REPORT------
868             %----------------------
869             %Create 'Log_Files' subfolder in 'OUTPUT_FILES'
870             LogFolder = fullfile(app.inputtxt_directory0, 'Log_Files');
871             if ~exist(LogFolder, 'dir')
872                 mkdir(LogFolder);
873             end
874             %Create the text file
875             filePath = fullfile(LogFolder, 'Log_Report_WaveHeightMap.txt');
876             fileId = fopen(filePath ,'w');
877
878             % Template for the default section of the log report (includes
information from the 'Input Data' section)
879             headerTitle = '******************** LOG REPORT
******************** ';
880             timestamp = datestr(now, 'yyyy-mm-dd HH:MM:SS');
881             endHeaderLine =
'************************************************';
882             footerSeparator =
'________________________________________________';
883             logContent = [
884                 sprintf('%s\n', headerTitle), ...
885                 sprintf('%-30s%-s\n', 'Map Type:', 'Wave Height Map'), ...
886                 sprintf('%-30s%-s\n', 'Timestamp:', timestamp), ...
887                 sprintf('%s\n\n', endHeaderLine), ...
888                 sprintf('%-30s%-s\n', 'Southwest Corner:', sprintf('Long: %f
Lat: %f', app.LongitudeEditField.Value, app.LatitudeEditField.Value)), ...
889                 sprintf('%-30s%-s\n', 'Grid Size:', sprintf('x: %d y: %d',
app.gridX.Value, app.gridY.Value)), ...
890                 sprintf('%-30s%-s\n', 'Simulation Time Start:', sprintf('%d
sec', app.StartTime.Value)), ...
891                 sprintf('%-30s%-s\n', 'Simulation Time Interval:', sprintf('%d
sec', app.TotalSimuilationTimesecEditField_3.Value)), ...
892                 sprintf('%s\n\n', footerSeparator)
893                 ];
894             fprintf(fileId, '%s', logContent);
895
896
897             %------LOG REPORT--------------------
898             %------Input Data: Import Files------
899             %----------------------------------
900             %Check the type of output file loaded in the app
901             patterns = {'hmax_', 'eta_','time_'};
902             matches1 = false(size(string(app.FileNAME)));
903             nonMatchingFiles = ~contains(string(app.FileNAME), patterns);
904             for i = 1:numel(patterns)
905                 pattern = patterns{i};
906                 matches1 = matches1 | contains(string(app.FileNAME), pattern);
907             end
908
909             if any(nonMatchingFiles)
910                 if contains(app.FileTextArea.Value, "eta_xxxx")
911                     fprintf(fileId, '%-30s%-s\n\n', 'IMPORT FILE/S:','Empty.
No basemap is plotted'); %Add an error note in the log report
912                 else
913                     % if (app.FileTextArea.Value ~= "0") % If non- eta/hmax
file is loaded
```

```
 914                     fprintf(fileId, '%-30s%-s\n', 'IMPORT FILE/S:', 'Use↙
filenames with ''eta_'', ''hmax_'' initials');
 915                     fprintf(fileId, '%-30s%-s\n', ' ' , 'Incorrect files are↙
loaded:');
 916                     fprintf(fileId, '%-30s%-s\n\n', ' ', join(app.FileNAME↙
(nonMatchingFiles), ', '));% List all filenames with errors in the log report
 917
 918                 end
 919             end
 920
 921             %------LOG REPORT-----------------------
 922             %------Input Data: Southwest Corner------
 923             %---------------------------------------
 924             %Check if the values have not changed and add an error note to the↙
log report
 925             if app.LongitudeEditField.Value == 0 && app.LatitudeEditField.↙
Value == 0
 926                 fprintf(fileId, '%-30s%-s\n\n', 'SOUTHWEST CORNER:','Both↙
latitude and longitude values are zeroes');
 927             elseif app.LongitudeEditField.Value == 0 && app.LatitudeEditField.↙
Value ~= 0
 928                 fprintf(fileId, '%-30s%-s\n\n', 'SOUTHWEST CORNER:','A default↙
zero value is used for longitude');
 929             elseif app.LongitudeEditField.Value ~= 0 && app.LatitudeEditField.↙
Value == 0
 930                 fprintf(fileId, '%-30s%-s\n\n', 'SOUTHWEST CORNER:','A default↙
zero value is used for latitude');
 931             end
 932
 933
 934             %------LOG REPORT-----------------
 935             %------Input Data: Grid Size------
 936             %---------------------------------
 937             %Check if the values have not changed and add an error note to the↙
log report
 938             if app.gridX.Value == 0 && app.gridY.Value == 0
 939                 fprintf(fileId, '%-30s%-s\n\n', 'GRID SIZE:','Both x and y↙
values are zeroes');
 940             elseif app.gridX.Value == 0 && app.gridY.Value ~= 0
 941                 fprintf(fileId, '%-30s%-s\n\n', 'GRID SIZE:','X value is↙
zero');
 942             elseif app.gridX.Value ~= 0 && app.gridY.Value == 0
 943                 fprintf(fileId, '%-30s%-s\n\n', 'GRID SIZE:','Y value is↙
zero');
 944             end
 945
 946             %------LOG REPORT-----------------------
 947             %------Input Data: Load Bathymetry------
 948             %---------------------------------------
 949             %Check if the matrix size of the loaded bathymetry file matches↙
the imported eta/hmax file
 950             if matches(app.FileTextArea.Value,string(app.FileNAME)) % Check if↙
the imported eta/hmax files match the names in the textbox
 951                 if matches(app.DepthFileEditField.Value,string(app.↙
bathymetryname)) %Check if the imported bathymetry matches the name in the textbox
 952                     if app.rowFirstFile ~= app.bathyRow
 953                         fprintf(fileId, '%-30s%-s\n', 'MATRIX ROWS:','Rows in↙
eta/hmax/time/ file and bathymetry do not match');
```

```
954                     end
955                     if app.colFirstFile ~= app.bathyCol
956                         fprintf(fileId, '%-30s%-s\n', 'MATRIX↙
COLUMNS:','Columns in eta/hmax/time file and bathymetry do not match');
957                     end
958                 end
959             end
960             if matches(app.DepthFileEditField.Value,"0") || ~matches(app.↙
DepthFileEditField.Value,string(app.bathymetryname)) % Add a log note that no↙
bathymetry is loaded, so the features to edit land area and bathymetry contours are↙
unavailable.
961                 fprintf(fileId, '%-30s%-s\n\n', 'BATHYMETRY:', 'Empty. Land↙
color and bathymetry contours are not applicable.');
962                 app.DepthFileEditField.FontColor = 'w';
963                 app.DepthFileEditField.BackgroundColor = 'r';
964                 app.DepthFileEditField.Value = 'NO FILE';
965             end
966
967
968             %------LOG REPORT-----------------
969             %------Basemap: Wave heights------
970             %---------------------------------
971             if all(contains(string(app.FileNAME),  patterns)) %If all files↙
are in eta_ /hmax_ format
972                 %Verify that the minimum, interval, and maximum input values↙
are correct; update values if conditions are not met
973                 %hmax option
974                 if app.MinEditField.Value > app.toEditField.Value
975                         temp = app.MinEditField.Value;
976                         app.MinEditField.Value = app.toEditField.Value;
977                         app.toEditField.Value = temp;
978                         drawnow
979                     fprintf(fileId, '%-30s%-s\n', 'BASEMAP HMAX↙
COLORBAR:','Max value must exceed min value');
980                     fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved: Min value↙
adjusted to load the map');
981                 end
982                 %eta option
983                 if app.MinEditField_2.Value > app.MaxEditField_2.Value
984                     temp = app.MinEditField_2.Value;
985                     app.MinEditField_2.Value = app.MaxEditField_2.Value;
986                     app.MaxEditField_2.Value = temp;
987                     drawnow
988                     fprintf(fileId, '%-30s%-s\n', 'BASEMAP ETA COLORBAR:','Max↙
value must exceed min value');
989                     fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved: Min value↙
adjusted to load the map');
990                 end
991             end
992
993             %------LOG REPORT---------------------------------
994             %------Overlay Features: Bathymetry contours------
995             %-------------------------------------------------
996             if matches(app.DepthFileEditField.Value,string(app.↙
bathymetryname))
997                 if app.MaximumEditField.Value < app.MinimumEditField.Value %↙
Update the max value if it is less than the user-defined minimum value.
998                     app.MaximumEditField.Value = 10000;
```

```
 999                     fprintf(fileId, '%-30s%-s\n', 'BATHYMETRY CONTOURS:','Max value must exceed min value');
1000                     fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved: Max value set to 10,000');
1001                 end
1002                 if app.IntervalEditField_5.Value > (app.MaximumEditField.Value -  app.MinimumEditField.Value) %Update the contour interval value when the condition is not met
1003                     app.IntervalEditField_5.Value = (app.MaximumEditField.Value -  app.MinimumEditField.Value)/2;
1004                     fprintf(fileId, '%-30s%-s\n', 'BATHYMETRY CONTOUR INTERVAL:','Value must be less than the max-min difference');
1005                     fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved: Value adjusted to half the max-min difference');
1006                 end
1007
1008                 if app.AddLabelsCheckBox_2.Value %Add labels
1009                     %Check if the label interval is divisible by the value provided in the 'Range' section
1010                     if mod(app.IntervalEditField_3.Value, app.IntervalEditField_5.Value) ~= 0
1011                         app.IntervalEditField_3.Value = app.IntervalEditField_5.Value;
1012                         fprintf(fileId, '%-30s%-s\n', 'BATHYMETRY CONTOUR LABEL:','Value must be divisible by the Interval value in the Range section');
1013                         fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved: Value adjusted to be the same as the Interval Value in the Range section');
1014                     end
1015
1016                     %Check if the label's interval value is less than the maximum value in 'Contour Range'
1017                     if app.IntervalEditField_3.Value > app.MaximumEditField.Value
1018                         app.IntervalEditField_3.Value = app.IntervalEditField_5.Value;
1019                         fprintf(fileId, '%-30s%-s\n', 'BATHYMETRY CONTOUR LABEL:','Value must be less than the maximum value of the contour line assigned in the Range section');
1020                         fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved: Value adjusted to be the same as the Interval Value in the Range section');
1021                     end
1022                 end
1023             end
1024
1025             if app.BathymetryButton_2.Value && (strcmp(app.DepthFileEditField.Value, "0") || strcmp(app.DepthFileEditField.Value, "00"))
1026                 %No bathymetry file found, show "NO FILE" warning in the bathymetry input window
1027                 app.DepthFileEditField.Value = "NO FILE";
1028                 app.DepthFileEditField.FontColor = 'w';
1029                 app.DepthFileEditField.BackgroundColor = 'r';
1030             end
1031
1032             %------LOG REPORT----------------------------------
1033             %------Overlay Features: Wave height contours------
1034             %--------------------------------------------------
1035             if app.MaximumdepthEditField_2.Value < app.MinimumdepthEditField 2.Value
```

```
1036                 app.MaximumdepthEditField_2.Value = 1;
1037                 app.MinimumdepthEditField_2.Value = 0;
1038                 fprintf(fileId, '%-30s%-s\n', 'WAVE HEIGHT CONTOURS:','Max value must exceed min value');
1039                 fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved: Max and min values reset to default [0,1]');
1040             end
1041
1042             if app.LineintervalEditField_2.Value  >  app.MaximumdepthEditField_2.Value - app.MinimumdepthEditField_2.Value
1043                 app.LineintervalEditField_2.Value = (app.MaximumdepthEditField_2.Value - app.MinimumdepthEditField_2.Value)/2;
1044                 fprintf(fileId, '%-30s%-s\n', 'WAVE HEIGHT CONTOUR INTERVAL:','Value must be less than the max-min difference');
1045                 fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved: Value adjusted to half the max-min difference');
1046             end
1047
1048             if app.AddLabelsCheckBox_3.Value %Add labels
1049                 if mod(app.wh_interval.Value, app.LineintervalEditField_2.Value) ~= 0
1050                     app.wh_interval.Value = app.LineintervalEditField_2.Value;
1051                     fprintf(fileId, '%-30s%-s\n', 'WAVE HEIGHT CONTOUR LABEL:','Value must be divisible by the Interval value in the Range section');
1052                     fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved: Value adjusted to be the same as the Interval Value in the Range section');
1053                 end
1054
1055                 %Check if the label's interval value is less than the maximum value in 'Contour Range'
1056                 if app.wh_interval.Value > app.MaximumdepthEditField_2.Value
1057                     app.wh_interval.Value = app.LineintervalEditField_2.Value;
1058                     fprintf(fileId, '%-30s%-s\n', 'WAVE HEIGHT CONTOUR LABEL:','Value must be less than the maximum value of the contour line assigned in the Range section');
1059                     fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved: Value adjusted to be the same as the Interval Value in the Range section');
1060                 end
1061             end
1062
1063             %------LOG REPORT--------------------------
1064             %------Overlay Features: Arrival Time------
1065             %------------------------------------------
1066             if app.ArrivalTimeButton.Value
1067                 if ~matches(app.FileEditField_3.Value, string(app.ArrivalTimename)) ||  matches(app.FileEditField_3.Value,"0")%Check if there is an uploaded file
1068                     fprintf(fileId, '%-30s%-s\n\n', 'ARRIVAL TIME:', 'Empty. No Arrival Time Contour plotted');
1069                     app.FileEditField_3.FontColor = 'w';
1070                     app.FileEditField_3.BackgroundColor = 'r';
1071                     app.FileEditField_3.Value = 'NO FILE';
1072                 end
1073                 if app.MinimumdepthEditField_3.Value > app.MaximumdepthEditField_3.Value
1074                     % Swap the values
1075                     tempArrivalTime = app.MinimumdepthEditField_3.Value;
1076                     app.MinimumdepthEditField_3.Value = app.
```

```
MaximumdepthEditField_3.Value;
1077                    app.MaximumdepthEditField_3.Value = tempArrivalTime;
1078                    fprintf(fileId, '%-30s%-s\n', 'ARRIVAL TIME:', 'Max value must exceed min value');
1079                    fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved: Values are swapped');
1080                end
1081                if app.LineintervalEditField_3.Value > app.MaximumdepthEditField_3.Value - app.MinimumdepthEditField_3.Value
1082                    app.LineintervalEditField_3.Value  = (app.MaximumdepthEditField_3.Value - app.MinimumdepthEditField_3.Value)/2;
1083                    fprintf(fileId, '%-30s%-s\n', 'ARRIVAL TIME INTERVAL:','Value must be less than the max-min difference');
1084                    fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved: Value adjusted to half the max-min difference');
1085                end
1086
1087                if app.AddLabelsCheckBox_5.Value %Add Labels
1088                    if mod(app.IntervalEditField_7.Value, app.LineintervalEditField_3.Value) ~= 0
1089                        app.IntervalEditField_7.Value = app.LineintervalEditField_3.Value;
1090                        fprintf(fileId, '%-30s%-s\n', 'ARRIVAL TIME LABEL:','Value must be divisible by the Interval value in the Range section');
1091                        fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved: Value adjusted to be the same as the Interval Value in the Range section');
1092                    end
1093
1094                    %Check if the label's interval value is less than the maximum value in 'Contour Range'
1095                    if app.IntervalEditField_7.Value > app.MaximumdepthEditField_3.Value
1096                        app.IntervalEditField_7.Value = app.LineintervalEditField_3.Value;
1097                        fprintf(fileId, '%-30s%-s\n', 'ARRIVAL TIME LABEL:','Value must be less than the maximum value of the contour line assigned in the Range section');
1098                        fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved: Value adjusted to be the same as the Interval Value in the Range section');
1099                    end
1100                end
1101            end
1102
1103
1104            %------LOG REPORT--------------------
1105            %------Overlay Features: Gauges------
1106            %------------------------------------
1107            if app.GaugesCheckBox.Value && ~matches(app.FileEditField.Value ,string(app.GAUGEFILE))
1108                fprintf(fileId, '%-30s%-s\n\n', 'GAUGES:', 'Empty. No virtual gauges plotted');
1109                app.FileEditField.Value = "NO FILE";
1110                app.FileEditField.FontColor = 'w';
1111                app.FileEditField.BackgroundColor = 'r';
1112            end
1113
1114            %------LOG REPORT--------------------------
1115            %------General Layout: Boundary Limits------
```

```
1116            %-----------------------------------------
1117            if app.WestEditField.Value > app.EastEditField.Value
1118                fprintf(fileId, '%-30s%-s\n', 'BOUNDARY LIMIT:','East boundary must be greater than West boundary');
1119                fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved: Values are swapped');
1120            elseif app.EastEditField.Value == app.WestEditField.Value
1121                fprintf(fileId, '%-30s%-s\n', 'BOUNDARY LIMIT:', 'East and West values are the same');
1122                fprintf(fileId, '%-30s%-s\n\n', 'Resolved:', 'The default values are used');
1123            end
1124
1125            if app.SouthEditField.Value > app.NorthEditField.Value
1126                fprintf(fileId, '%-30s%-s\n', 'BOUNDARY LIMIT:','North boundary must be greater than South boundary');
1127                fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved: Values are swapped');
1128            elseif  app.NorthEditField.Value == app.SouthEditField.Value
1129                fprintf(fileId, '%-30s%-s\n', 'BOUNDARY LIMIT:', 'North and South values are the same');
1130                fprintf(fileId, '%-30s%-s\n\n', 'Resolved:', 'The default values are used');
1131            end
1132
1133            %---------------------------
1134            %------SAVE MAP: .tif-------
1135            %---------------------------
1136            if app.tifCheckBox_2.Value && (app.LatitudeEditField.Value > 90 || app.LatitudeEditField.Value < -90)
1137                fprintf(fileId, '%-30s%-s\n\n', 'BOUNDARY LIMIT:','The Latitude values should be within -90 to 90.');
1138            end
1139
1140            fclose(fileId); %Close the log report file
1141
1142            %---------------------------------
1143            %------PRE-SET MAP OPTIONS-------
1144            %---------------------------------
1145            clear depth f1
1146            FileLength = length(app.FileINPUT); %Read all the imported eta/hmax files
1147
1148            %------Plot all data in one figure------
1149            if app.PlotalldatainonefigureButton.Value
1150                %Create the figure
1151                figure(1);
1152
1153                %Formatting the subplots
1154                if FileLength == 1
1155                    plotCol = 1;
1156                    plotRow = 1;
1157                elseif FileLength > 1 && FileLength <= 10
1158                    plotCol = 2;
1159                    plotRow = ceil(FileLength/2);
1160                elseif FileLength > 10
1161                    plotCol = 3;
1162                    plotRow = ceil(FileLength/3);
```

```
1163                end
1164
1165            end
1166
1167            %-----Bathymetry data------
1168            % bathymetryFileExists = exist(app.bathymetryinputdata, 'file') ==↙
2;
1169            % if bathymetryFileExists
1170                depth = app.bathymetrydata;
1171
1172                % Flip the data horizontally
1173                if app.HorizontalCheckBox.Value
1174                    depth = fliplr(depth);
1175                end
1176
1177                % Flip the data vertically
1178                if app.VerticalCheckBox.Value
1179                    depth = flipud(depth);
1180                end
1181            % end
1182
1183
1184            %------Basemap Colormap-------
1185            colorbarHandles = [];
1186            if strcmp(datacolorbrewerETA,'none') %Using default matlab↙
colormap options
1187                datacolor2ETA = evalin('base',datacolorETA);
1188            else %Using cbrewer2
1189                datacolor2ETA = cbrewer2(datacolorbrewerETA,datacolorETA,app.↙
ColorinterpolationEditField_2.Value,'pchip');
1190            end
1191
1192            if strcmp(datacolorbrewer,'none')  %Using default matlab colormap↙
options
1193                datacolor2 =  evalin('base',datacolor);
1194            else %Using cbrewer2
1195                datacolor2 = cbrewer2(datacolorbrewer,datacolor,app.↙
DivisionEditField.Value,'pchip');
1196            end
1197
1198            colormapMinHMAX = app.MinEditField.Value;
1199            colormapMaxHMAX = app.toEditField.Value;
1200            colorbartitleHMAX = 'Maximum \eta (m)';
1201
1202            colormapMinETA = app.MinEditField_2.Value;
1203            colormapMaxETA = app.MaxEditField_2.Value;
1204            colorbartitleETA = '\eta (m)';
1205
1206            colorbarFontSize = app.ColorbarTextSize.Value;
1207            flipCheckBoxETA = app.FlipCheckBox_2.Value;
1208            flipCheckBoxHMAX = app.FlipCheckBox_5.Value;
1209
1210            %------Bathymetry Contours------
1211            if app.BathymetryButton_2.Value && matches(app.DepthFileEditField.↙
Value,string(app.bathymetryname))
1212                kernel = ones(6) / 36;  %Make the contour smoother
1213                wavebathymetry=  filter2(kernel, depth);
1214                contourLevels = app.MinimumEditField.Value:app.↙
```

```
IntervalEditField_5.Value:app.MaximumEditField.Value;
1215                 manualLabels = app.MinimumEditField.Value:app.↙
IntervalEditField_3.Value:app.MaximumEditField.Value;
1216             end
1217
1218
1219             isBathymetryChecked = app.BathymetryButton_2.Value;
1220             isMatchingBathymetry = matches(app.DepthFileEditField.Value,↙
string(app.bathymetryname));
1221             isAddingLabels = app.AddLabelsCheckBox_2.Value;
1222             fontSize = app.TextLabelSize.Value;
1223             labelSpacing = app.LabelSpacingEditField.Value;
1224             lineWidth = app.WidthEditField.Value;
1225             contourCol = ContourColor; % Capture contour color if it's same↙
for all
1226             contourStyle = ContourLineStyle; % Capture contour style if it's↙
same for all
1227
1228
1229             %------Wave Height Contours------
1230             if app.WaveHeightButton.Value
1231                 wavecontour = app.fin;
1232                 if exist('app.bathymetrydata', 'var')
1233                     wavecontour(depth<=0) = NaN; %Remove values on land
1234                 end
1235                 contourLevelsWH = app.MinimumdepthEditField_2.Value:app.↙
LineintervalEditField_2.Value:app.MaximumdepthEditField_2.Value;
1236                 manualLabels2 = app.MinimumdepthEditField_2.Value:app.↙
wh_interval.Value:app.MaximumdepthEditField_2.Value;
1237             end
1238
1239             waveContourColor = WaveContourColor;
1240             waveContourStyle = WaveContourLineStyle;
1241             waveLineWidth = app.ThicknessEditField_2.Value;
1242             fontSize2 = app.TextLabelSize_2.Value;
1243             labelSpacing2 = app.LabelSpacingEditField_2.Value;
1244
1245             %------Arrival Time Contours------
1246             if app.ArrivalTimeButton.Value && matches(app.FileEditField_3.↙
Value, string(app.ArrivalTimename))
1247                 atc = app.ArrivalTimeConverted;
1248
1249                 %Flip the data vertically
1250                 if app.VerticalCheckBox.Value
1251                     atc = flipud(app.ArrivalTimeConverted);
1252                 end
1253
1254                 %Flip the data horizontally
1255                 if app.HorizontalCheckBox.Value
1256                     atc = fliplr(app.ArrivalTimeConverted);
1257                 end
1258
1259                 contourLevelsATC = app.MinimumdepthEditField_3.Value:app.↙
LineintervalEditField_3.Value:app.MaximumdepthEditField_3.Value;
1260                 manualLabels3 = app.MinimumdepthEditField_3.Value:app.↙
IntervalEditField_7.Value:app.MaximumdepthEditField_3.Value;
1261             end
1262
```

```
1263            atcContourColor = ArrivalTimeContourColor;
1264            atcContourStyle = ArrivalTimeLineStyle;
1265            atcLineWidth = app.ThicknessEditField_3.Value;
1266            fontSize3 = app.TextLabelSize_3.Value;
1267            labelSpacing3 = app.LabelSpacingEditField_4.Value;
1268
1269            %------Gauges------
1270            markerSize = app.SizeEditField.Value;
1271            labelFontSize = app.SizeEditField_3.Value;
1272
1273            %------Boundary limit-------
1274            eastValue = app.EastEditField.Value;
1275            westValue = app.WestEditField.Value;
1276
1277            if eastValue < westValue
1278                % Swap West and East input values if the West value is greater ↙
than the East
1279                xMin_WH = eastValue;
1280                xMax_WH = westValue;
1281                app.EastEditField.Value = xMax_WH;
1282                app.WestEditField.Value = xMin_WH;
1283            elseif eastValue == westValue
1284                % Set the limits to the default values
1285                app.EastEditField.Value = max(app.x);
1286                app.WestEditField.Value = min(app.x);
1287                xMin_WH = app.WestEditField.Value;
1288                xMax_WH = app.EastEditField.Value;
1289            else
1290                xMin_WH = westValue;
1291                xMax_WH = eastValue;
1292            end
1293
1294            northValue = app.NorthEditField.Value;
1295            southValue = app.SouthEditField.Value;
1296
1297            if northValue < southValue
1298                % Swap North and South input values if the South value is ↙
greater than the North
1299                yMin_WH = northValue;
1300                yMax_WH = southValue;
1301                app.NorthEditField.Value = yMax_WH;
1302                app.SouthEditField.Value = yMin_WH;
1303            elseif northValue == southValue
1304                % Set the limits to the default values
1305                app.NorthEditField.Value = max(app.y);
1306                app.SouthEditField.Value = min(app.y);
1307            else
1308                yMin_WH = southValue;
1309                yMax_WH = northValue;
1310            end
1311
1312            %------Map Title-------
1313            totalSimulationTime = app.TotalSimuilationTimesecEditField_3. ↙
Value;
1314            startTime = app.StartTime.Value;
1315            colorbarTextSize = app.ColorbarTextSize.Value + 1;
1316            pats = digitsPattern;
1317
```

```
1318
1319
1320            %-----------------------------------------
1321            %------Start of the loop for plotting-------
1322            %-----------------------------------------
1323            for i = 1:FileLength
1324                if all(contains(string(app.FileNAME), patterns)) % Check if each file is either an eta or hmax file
1325                    f1 = readmatrix(app.FileINPUT(i));
1326
1327                    % Flip the data horizontally
1328                    if app.HorizontalCheckBox.Value
1329                        f1 = fliplr(f1);
1330                    end
1331
1332                    % Flip the data vertically
1333                    if app.VerticalCheckBox.Value
1334                        f1 = flipud(f1);
1335                    end
1336
1337                    if matches(app.DepthFileEditField.Value,string(app.bathymetryname))
1338                        %If the bathymetry file is loaded, delete negative land values
1339                        f1(depth <= 0) = NaN;
1340                    end
1341
1342                    % if matches(app.DepthFileEditField.Value, string(app.bathymetryname))
1343                    %     % Ensure depth is defined and non-empty
1344                    %     if ~isempty(depth)
1345                    %         % Preallocate f1 to match the size of depth
1346                    %         f1 = NaN(size(depth));  % Initialize with NaNs
1347                    %         % Optionally assign valid values where depth > 0
1348                    %         % f1(depth > 0) = someComputation(depth(depth > 0));
1349                    %     % else
1350                    %     %     warning('Depth data is empty. Skipping bathymetry filtering.');
1351                    %     end
1352                    % end
1353                    app.fin = f1;
1354
1355                    %----------------------------------------
1356                    %------PLOT ALL DATA IN ONE FIGURE------
1357                    %----------------------------------------
1358                    if app.PlotalldatainonefigureButton.Value
1359                        gca(i)= subplot(plotRow,plotCol,i);
1360                        %--------------------------
1361                        %------PLOT SEPARATELY------
1362                        %--------------------------
1363                    elseif app.PlotseparatelyButton_2.Value
1364                        figure();
1365                        gca(i) = subplot(1,1,1);
1366                    end
1367
1368                    %---------------------
1369                    %------MAIN PLOT------
```

```
1370                    %---------------------
1371
1372                    f = pcolor(app.x,  app.y, (app.fin)); %(X coordinates, X↙
coordinates, eta/hmax values)
1373                    shading interp;
1374                    set(f, 'EdgeColor', 'none', 'FaceColor', 'interp');
1375                    set(gca,'Color', app.LandAreaColor) %Set the background↙
color for the land
1376                    hold on
1377
1378
1379                    %------------------------------
1380                    %------BASEMAP: Colormap-------
1381                    %------------------------------
1382                    cb = colorbar;
1383                    colorbarHandles = [colorbarHandles, cb];
1384
1385                    % Set the properties once
1386                    cb.Label.FontSize = colorbarFontSize; % Colorbar label
1387                    cb.FontSize = colorbarFontSize;       % Size of the tick↙
number% Size of the tick number
1388
1389                    if contains(string(app.FileNAME(i)), "eta") % For ETA/sea↙
surface displacement at time t
1390                        % Flip the colorbar if needed
1391                        if flipCheckBoxETA
1392                            colormap(gca(i), flipud(datacolor2ETA));
1393                        else
1394                            colormap(gca(i), datacolor2ETA);
1395                        end
1396                        caxis(gca(i), [colormapMinETA, colormapMaxETA]); %↙
Colorbar min and max values
1397                        cb.Label.String = colorbartitleETA;
1398                    else % For hmax/maximum wave height
1399                        % Flip the colorbar if needed
1400                        if flipCheckBoxHMAX
1401                            colormap(gca(i), flipud(datacolor2));
1402                        else
1403                            colormap(gca(i), datacolor2);
1404                        end
1405                        caxis(gca(i), [colormapMinHMAX, colormapMaxHMAX]); %↙
Colorbar min and max values
1406                        cb.Label.String = colorbartitleHMAX;
1407                    end
1408
1409                    %---------------------------------------------------
1410                    %------OVERLAY FEATURES: Bathymetry Contours-------
1411                    %---------------------------------------------------
1412                    if isBathymetryChecked && isMatchingBathymetry
1413                        %                           hold on
1414                        % Plot the contours
1415                        [C, h] = contour(app.x, app.y, wavebathymetry,↙
contourLevels, 'EdgeColor', contourCol, 'LineStyle', contourStyle, 'LineWidth',↙
lineWidth);
1416                        % Add contour labels if needed
1417                        if isAddingLabels
1418                            clabel(C, h, manualLabels, 'FontSize', fontSize,↙
'Color', 'k');
```

```
1419                             set(findobj(h, '-property', 'LabelSpacing'),↙
'LabelSpacing', labelSpacing); % Adjust the label spacing
1420                         end
1421                     end
1422
1423                     %-----------------------------------------------
1424                     %------OVERLAY FEATURES: Wave Height Contours-------
1425                     %-----------------------------------------------
1426                     if app.WaveHeightButton.Value
1427                         % Plot the contours
1428                         [D, g] = contour(app.x, app.y, wavecontour,↙
contourLevelsWH, 'EdgeColor', waveContourColor, 'LineStyle', waveContourStyle,↙
'LineWidth', waveLineWidth);
1429
1430                         % Add contour labels
1431                         if app.AddLabelsCheckBox_3.Value
1432                             clabel(D, g, manualLabels2, 'FontSize', fontSize2,↙
'Color', 'k');
1433                             set(findobj(g, '-property', 'LabelSpacing'),↙
'LabelSpacing', labelSpacing2); % Adjust the label spacing
1434                         end
1435                     end
1436
1437                     %-----------------------------------------------
1438                     %------OVERLAY FEATURES: Arrival Time Contours-------
1439                     %-----------------------------------------------
1440                     if app.ArrivalTimeButton.Value && matches(app.↙
FileEditField_3.Value, string(app.ArrivalTimename))
1441                         % Plot the contours
1442                         [E, ggg] = contour(app.x, app.y, atc,↙
contourLevelsATC, 'EdgeColor', atcContourColor, 'LineStyle', atcContourStyle,↙
'LineWidth', atcLineWidth);
1443
1444                         if app.AddLabelsCheckBox_5.Value
1445                             clabel(E, ggg, manualLabels3, 'FontSize',↙
fontSize3, 'Color', 'k');
1446                             set(findobj(ggg, '-property', 'LabelSpacing'),↙
'LabelSpacing', labelSpacing3); % Adjust the label spacing
1447                         end
1448                     end
1449
1450                     %---------------------------------------
1451                     %------OVERLAY FEATURES: Gauges-------
1452                     %---------------------------------------
1453                     if app.GaugesCheckBox.Value && matches(app.FileEditField.↙
Value, string(app.GAUGEFILE))
1454                         % Plot the points
1455                         gaugePoint = plot(app.longGauge, app.latGauge,↙
GaugeMarker);
1456                         set(gaugePoint, 'MarkerSize', markerSize,↙
'MarkerFaceColor', GaugeColor, 'MarkerEdgeColor', GaugeColor); % Size and color of↙
the points
1457
1458                         % Add labels
1459                         if app.AddLabelsCheckBox_4.Value
1460                             horzLabel = HorzLabel; % Ensure HorzLabel is↙
precomputed
1461                             vertLabel = VertLabel; % Ensure VertLabel is↙
```

```
precomputed
1462                                gSpacing = app.GLspacing.Value;
1463
1464                                for k = 1:length(app.latGauge)
1465                                    gtxt = text(app.longGauge(k) + gSpacing, app.↙
latGauge(k), num2str(k), 'FontSize', labelFontSize, 'HorizontalAlignment',↙
horzLabel, 'VerticalAlignment', vertLabel);
1466                                    set(gtxt, 'Clipping', 'on');
1467                                end
1468                            end
1469                        end
1470                        hold on
1471
1472                        %-----------------------------------------
1473                        %------GENERAL LAYOUT: Boundary limit-------
1474                        %-----------------------------------------
1475                        xlim([xMin_WH,xMax_WH])
1476                        ylim([yMin_WH,yMax_WH])
1477                        hold on
1478
1479                            %---------------------------------
1480                            %------GENERAL LAYOUT: Ticks-------
1481                            %---------------------------------
1482                            %Reduce the number of ticks to 2 on the X axis and 2↙
on the Y axis for simplicity
1483                            % Get the current ticks from the first plot
1484                            if i == 1
1485                                xAllTicks = xticks(gca);
1486                                yAllTicks = yticks(gca);
1487
1488                                % Ensure the ticks are numeric arrays
1489                                if iscell(xAllTicks)
1490                                    xAllTicks = cellfun(@str2double, xAllTicks,↙
'UniformOutput', true);
1491                                end
1492                                if iscell(yAllTicks)
1493                                    yAllTicks = cellfun(@str2double, yAllTicks,↙
'UniformOutput', true);
1494                                end
1495
1496                                % Remove any NaN values that might result from↙
str2double conversion
1497                                xAllTicks = xAllTicks(~isnan(xAllTicks));
1498                                yAllTicks = yAllTicks(~isnan(yAllTicks));
1499
1500                                % Select only the 2nd and second-to-last ticks
1501                                % For X axis
1502                                if length(xAllTicks) >= 4  % Check first if there↙
are enough ticks to select from
1503                                    xticksSelected = sort([xAllTicks(2), xAllTicks↙
(end-1)]);
1504                                elseif length(xAllTicks) >= 2
1505                                    xticksSelected = sort([xAllTicks(1), xAllTicks↙
(end)]); % Use the first and last if less than 4
1506                                else
1507                                    xticksSelected = xAllTicks; % Use whatever↙
ticks are available
1508                                end
```

```
1509
1510                                % For Y axis
1511                                if length(yAllTicks) >= 4
1512                                    yticksSelected = sort([yAllTicks(2), yAllTicks↙
(end-1)]);
1513                                elseif length(yAllTicks) >= 2
1514                                    yticksSelected = sort([yAllTicks(1), yAllTicks↙
(end)]); % Use the first and last if less than 4
1515                                else
1516                                    yticksSelected = yAllTicks; % Use whatever↙
ticks are available
1517                                end
1518
1519                            end
1520
1521                            % Update the number of ticks on the axes
1522                            xticks(gca, xticksSelected);
1523                            yticks(gca, yticksSelected);
1524
1525                            % Determine the maximum decimal places for x and y↙
ticks
1526                            xmaxDecimals = max(cellfun(@(x) length(regexp(x, '(?↙
<=\.)\d+', 'match', 'once')), ...
1527                                cellstr(num2str(xticksSelected'))));
1528                            ymaxDecimals = max(cellfun(@(y) length(regexp(y, '(?↙
<=\.)\d+', 'match', 'once')), ...
1529                                cellstr(num2str(yticksSelected'))));
1530
1531                            % Convert tick labels to their equivalent in degrees
1532                            if app.PlotindegreesCheckBox_2.Value
1533                                % Convert selected ticks to formatted strings and↙
set them for degrees
1534                                xticklabels(gca, arrayfun(@(x) sprintf('%.1f°',↙
mod(x + 360, 360)), xticksSelected, 'UniformOutput', false));
1535                                yticklabels(gca, arrayfun(@(y) sprintf('%.1f°',↙
y), yticksSelected, 'UniformOutput', false));
1536                            else
1537                                % Correct the format string and apply calculated↙
maximum decimals
1538                                xticklabels(gca, arrayfun(@(x) sprintf(['%.'↙
num2str(xmaxDecimals) 'f'], x), xticksSelected, 'UniformOutput', false));
1539                                yticklabels(gca, arrayfun(@(y) sprintf(['%.'↙
num2str(ymaxDecimals) 'f'], y), yticksSelected, 'UniformOutput', false));
1540                            end
1541
1542                            % Set the y-axis tick angle
1543                            ytickangle(gca, -270);
1544
1545                            %Fontsize of the tick labels
1546                            set(gca,'Layer', 'top','LineWidth',1,'FontSize',app.↙
ColorbarTextSize.Value-1)
1547
1548                            %Adjust tick lengths based on the number of subplots↙
if 'Plot all data in one figure' is selected
1549                            if FileLength <= 2 && app.↙
PlotalldatainonefigureButton.Value
1550                                set(gca,'TickLength',[0.015 0.015])
1551                            else
```

```
1552                                set(gca,'TickLength',[0.01 0.01])
1553                            end
1554
1555                            grid off
1556                            box on
1557                            daspect([1 1 1]) %check if it distorts the countries
1558                            hold off
1559
1560                            %-------------------------------------
1561                            %------GENERAL LAYOUT: Map Title------
1562                            %-------------------------------------
1563                            str = string(app.FileNAME(i));
1564                            dbl1 = extract(str, pats); % Extract numbers from the↙
eta/hmax filename to calculate the time frame
1565
1566                            if startTime == 0 && str2double(dbl1) == 0 %Set the↙
time to zero
1567                                timeT = startTime + (str2double(dbl1)-1);
1568                            else
1569                                originalTimeT = totalSimulationTime * (str2double↙
(dbl1)-1); % Convert the filenumber to time
1570                                timeT = startTime + originalTimeT; % Adjust the↙
start of the time count
1571                            end
1572
1573                            % Time conversion
1574                            if timeT < 60
1575                                timeStr = sprintf("%d sec", timeT);
1576                            elseif timeT < 3600
1577                                minutes = round(timeT / 60);
1578                                if minutes == 1
1579                                    timeStr = "1 min";
1580                                else
1581                                    timeStr = sprintf("%d mins", minutes);
1582                                end
1583                            else
1584                                raw_hours = timeT/3600;
1585                                hours = fix(timeT/3600); %Extract the whole↙
number%Extract the whole number
1586                                hours_decimal = raw_hours - floor(raw_hours); %↙
Extract the decimal places
1587                                minutes = round(hours_decimal *  60);
1588
1589                                if hours > 0
1590                                    if minutes > 0
1591                                        timeStr = sprintf("%d hr, %d mins", hours,↙
minutes);
1592                                    else
1593                                        timeStr = sprintf("%d hr", hours);
1594                                    end
1595                                end
1596                            end
1597
1598                            title(timeStr, 'FontSize', colorbarTextSize); % Add↙
the title to the top of the map; Adjust the font size of the title
1599
1600                            %-----------------------------------------
1601                            %------GENERAL LAYOUT: Figure Size-------
```

```
1602                            %---------------------------------------
1603                            if ~app.AutoSetCheckBox_2.Value
1604                                %Manually set the figure size
1605                                figureHandle = ancestor(f, 'figure'); %
1606                                set(figureHandle, 'Units', 'Inches', 'Position',↙
[0, 0, app.Width_2.Value, app.Height_2.Value], 'PaperUnits', 'Inches', 'PaperSize',↙
[app.Width_2.Value, app.Height_2.Value]);
1607                                set(gcf, 'Units', 'Inches', 'Position', [0, 0,↙
app.Width_2.Value, app.Height_2.Value], 'PaperUnits', 'Inches', 'PaperSize', [app.↙
Width_2.Value, app.Height_2.Value]);
1608                            end
1609
1610
1611                            %---------------------------------------
1612                            %------SAVE MAP: Plot Separately-------
1613                            %---------------------------------------
1614                            if app.PlotseparatelyButton_2.Value
1615                                %Save the maps with filenames matching the loaded↙
file
1616                                OF  = fullfile(app.inputtxt_directory1,app.↙
FileNAME(i));
1617
1618                                %Options for saving format
1619                                %PNG
1620                                if app.pngCheckBox.Value
1621                                    outputfile = OF + ".png";
1622                                    exportgraphics(gcf,outputfile,'Resolution',↙
300)
1623                                end
1624
1625                                %JPG
1626                                if app.jpgCheckBox.Value
1627                                    outputfile = OF + ".jpg";
1628                                    exportgraphics(gcf,outputfile,'Resolution',↙
300)
1629                                end
1630
1631                                %TIF
1632                                if app.tifCheckBox_2.Value
1633                                    XXX = app.x;
1634                                    YYY = app.y;
1635
1636                                    west_limit = app.WestEditField.Value;
1637                                    east_limit = app.EastEditField.Value;
1638                                    north_limit = app.NorthEditField.Value;
1639                                    south_limit = app.SouthEditField.Value;
1640
1641                                    %Make the longitude values within -180 to 180↙
range
1642                                    if west_limit ~= 180
1643                                        west_limit = mod(west_limit + 180, 360) -↙
180;
1644                                    end
1645
1646                                    if east_limit ~= 180
1647                                        east_limit = mod(east_limit + 180, 360) -↙
180;
1648                                    end
```

```
1649
1650                                % Convert x coordinates to correct degree↙
range
1651                                XXX(XXX > 180) = XXX(XXX > 180) - 360;
1652
1653                                % Create meshgrid
1654                                [X, Y] = meshgrid(XXX, YYY);
1655                                Z = f1;
1656                                %Check if the latitude is within the↙
Geographic limits
1657                                if app.LatitudeEditField.Value > 90 || app.↙
LatitudeEditField.Value < -90
1658                                    app.LatitudeEditField.Value = 0;
1659                                else
1660                                    %Clip the matrix based on boundary limits
1661                                    if west_limit >= 0 && east_limit >=0  ||↙
west_limit < 0 && east_limit < 0 % if east and west values are either both (+) or↙
both (-) values
1662                                        x_indices_positive =  XXX >=↙
west_limit & XXX <= east_limit;
1663                                        y_indices = YYY >= south_limit & YYY↙
<= north_limit;
1664                                        Z_positive = Z(y_indices,↙
x_indices_positive);
1665                                        X_positive = X(y_indices,↙
x_indices_positive);
1666                                        Y = Y(y_indices, x_indices_positive);↙
% Keep the same for both file
1667                                        if ~isempty(Z_positive)
1668                                            % Define spatial referencing↙
information
1669                                            R_positive = georasterref↙
('RasterSize', size(Z_positive), 'LatitudeLimits', [min(Y(:)), max(Y(:))],↙
'LongitudeLimits', [min(X_positive(:)), max(X_positive(:))]);
1670                                            R_positive.ColumnsStartFrom =↙
'south'; % Set the column orientation to start from the north
1671                                            R_positive.RowsStartFrom = 'west';
1672
1673                                            % Save as geotiff
1674                                            filename_positive = OF + ".tif";
1675                                            geotiffwrite(filename_positive,↙
Z_positive, R_positive);
1676                                        end
1677
1678                                    elseif west_limit >= 0 && east_limit < 0↙
%if longitude values are a combination of (+) west limit and (-) east limit
1679                                        x_indices_positive = XXX >= 0 & XXX >=↙
west_limit;
1680                                        x_indices_negative = XXX < 0 & XXX <=↙
east_limit;
1681                                        y_indices = YYY >= south_limit & YYY↙
<= north_limit;
1682                                        Z_positive = Z(y_indices,↙
x_indices_positive);
1683                                        Z_negative = Z(y_indices,↙
x_indices_negative);
1684                                        X_positive = X(y_indices,↙
x_indices_positive);
```

```
1685                                            X_negative = X(y_indices, x_indices_negative);
1686                                            Y = Y(y_indices, x_indices_positive); % Keep the same for both file
1687
1688                                            if ~isempty(Z_positive)
1689                                                % Define spatial referencing information
1690                                                R_positive = georasterref('RasterSize', size(Z_positive), 'LatitudeLimits', [min(Y(:)), max(Y(:))], 'LongitudeLimits', [min(X_positive(:)), max(X_positive(:))]);
1691                                                R_positive.ColumnsStartFrom = 'south'; % Set the column orientation to start from the north
1692                                                R_positive.RowsStartFrom = 'west';
1693
1694                                                % Save as geotiff
1695                                                filename_positive = OF + "_1.tif";
1696                                                geotiffwrite(filename_positive, Z_positive, R_positive);
1697                                            end
1698
1699                                            if ~isempty(Z_negative)
1700                                                % Define spatial referencing information
1701                                                R_negative = georasterref('RasterSize', size(Z_negative), 'LatitudeLimits', [min(Y(:)), max(Y(:))], 'LongitudeLimits', [min(X_negative(:)), max(X_negative(:))]);
1702                                                R_negative.ColumnsStartFrom = 'south'; % Set the column orientation to start from the north
1703                                                R_negative.RowsStartFrom = 'west';
1704
1705                                                % Save as geotiff
1706                                                filename_negative = OF + "_2.tif";
1707                                                geotiffwrite(filename_negative, Z_negative, R_negative)
1708                                            end
1709
1710                                        elseif west_limit < 0 && east_limit >= 0 %if longitude values are a combination of (-) west limit and (+) east limit
1711                                            x_indices_positive = XXX >= 0 & XXX <= east_limit;
1712                                            x_indices_negative = XXX < 0 & XXX >= west_limit;
1713                                            y_indices = YYY >= south_limit & YYY <= north_limit;
1714                                            Z_positive = Z(y_indices, x_indices_positive);
1715                                            Z_negative = Z(y_indices, x_indices_negative);
1716                                            X_positive = X(y_indices, x_indices_positive);
1717                                            X_negative = X(y_indices, x_indices_negative);
1718                                            Y = Y(y_indices, x_indices_positive); % Keep the same for both file
1719
1720                                            if ~isempty(Z_positive)
1721                                                % Define spatial referencing
```

```
information
                                                    R_positive = georasterref↙
('RasterSize', size(Z_positive), 'LatitudeLimits', [min(Y(:)), max(Y(:))],↙
'LongitudeLimits', [min(X_positive(:)), max(X_positive(:))]);
                                                    R_positive.ColumnsStartFrom =↙
'south'; % Set the column orientation to start from the north
                                                    R_positive.RowsStartFrom = 'west';

                                                    % Save as geotiff
                                                    filename_positive = OF + "_1.tif";
                                                    geotiffwrite(filename_positive,↙
Z_positive, R_positive);
                                                end
                                                if ~isempty(Z_negative)
                                                    % Define spatial referencing↙
information
                                                    R_negative = georasterref↙
('RasterSize', size(Z_negative), 'LatitudeLimits', [min(Y(:)), max(Y(:))],↙
'LongitudeLimits', [min(X_negative(:)), max(X_negative(:))]);
                                                    R_negative.ColumnsStartFrom =↙
'south'; % Set the column orientation to start from the north
                                                    R_negative.RowsStartFrom = 'west';

                                                    % Save as geotiff
                                                    filename_negative = OF + "_2.tif";
                                                    geotiffwrite(filename_negative,↙
Z_negative, R_negative);
                                                end
                                            end
                                        end
                                    end

                                    %Create animation / closing the file
                                    if app.mp4CheckBox.Value
                                        set(gcf,'Renderer','zbuffer');
                                        f = getframe(gcf);
                                        writeVideo(vidObj,f);
                                    end
                                end

                            end

                        end

                        %Close the animation
                        if app.mp4CheckBox.Value
                            close(vidObj);
                        end

                        %-------------------------------------------------
                        %------SAVE MAP: Plot all data in one figure-------
                        %-------------------------------------------------
                        if app.PlotalldatainonefigureButton.Value
                            %Set up the filename
                            if FileLength == 1 %Use raw hmax/eta filename
                                OF = fullfile(app.inputtxt_directory1,string(app.↙
FileNAME));
                            else %Use a generic name
```

```
1769                    OF = fullfile(app.inputtxt_directory1,↙
"WaveHeight_Output");
1770                end
1771
1772                %Save the figure
1773                if app.pngCheckBox.Value
1774                    outputfile = OF + ".png";
1775                    exportgraphics(gcf,outputfile,'Resolution',300)
1776                end
1777
1778                if app.jpgCheckBox.Value
1779                    outputfile = OF + ".jpg";
1780                    exportgraphics(gcf,outputfile,'Resolution',300)
1781                end
1782            end
1783
1784
1785
1786        end
1787
1788        % Button pushed function: Button_15
1789        function Button_15Pushed(app, event)
1790
1791
1792            %Load the bathymetry file and display its filename in the textbox
1793            [bathymetryfile,path2] = uigetfile('*', 'Select the files',↙
'MultiSelect', 'on');
1794            if ~(ismcc || isdeployed)
1795                addpath(genpath(fullfile(string(path2))));
1796            end
1797            app.DepthFileEditField.Value = string(bathymetryfile);
1798            app.bathymetryinputdata = string(path2) + string(bathymetryfile);
1799            app.bathymetryname = string(bathymetryfile);
1800            app.DepthFileEditField.BackgroundColor = 'w';
1801            app.DepthFileEditField.FontColor =  'k';
1802
1803            %Extract information from the bathymetry
1804            if app.DepthFileEditField.Value == string(bathymetryfile)
1805                if ~strcmp(app.DepthFileEditField.Value, "0") || ~strcmp(app.↙
DepthFileEditField.Value, "00")
1806                    % If the file is .tif, read the matrix, extract the↙
geographic values and grid sizes, and display them in the app textboxes
1807                    if contains(app.bathymetryinputdata,'.tif')
1808                        [A1,R2] = readgeoraster(app.bathymetryinputdata);
1809                        app.bathymetrydata = flipud(A1);
1810
1811                        %Extract lat and long of the southwest corner
1812                        app.LongitudeEditField.Value = min(R2.↙
LongitudeLimits);
1813                        app.LatitudeEditField.Value = min(R2.LatitudeLimits);
1814                        app.LongitudeEditField.FontColor = 'k';
1815                        app.LatitudeEditField.FontColor = 'k';
1816
1817                        %Extract grid size
1818                        app.gridX.Value = R2.CellExtentInLatitude;
1819                        app.gridY.Value = R2.CellExtentInLongitude;
1820                        app.gridX.FontColor = 'k';
1821                        app.gridY.FontColor = 'k';
```

```
1822                            drawnow
1823
1824                            %Update the X-axis limits in the 'General Layout' ↙
section
1825                            if isprop(app, 'col1')
1826                                app.x = app.LongitudeEditField.Value + [0:app. ↙
col1] * app.gridX.Value;
1827                                app.EastEditField.Value = max(app.x);
1828                                app.WestEditField.Value = min(app.x);
1829                                app.EastEditField.FontColor = 'k';
1830                                app.WestEditField.FontColor = 'k';
1831                            end
1832
1833                            %Update the Y-axis limits in the 'General Layout' ↙
section
1834                            if isprop(app, 'row1')
1835                                app.y = app.LatitudeEditField.Value + [0:app.row1] ↙
* app.gridY.Value;
1836                                app.NorthEditField.Value = max(app.y);
1837                                app.SouthEditField.Value = min(app.y);
1838                                app.NorthEditField.FontColor = 'k';
1839                                app.SouthEditField.FontColor = 'k';
1840                            end
1841                            drawnow
1842
1843                        else   %Only read the matrix for use as a basemap and for ↙
plotting bathymetry contours
1844                            app.bathymetrydata = readmatrix(app. ↙
bathymetryinputdata);
1845                            app.MaximumEditField.Value = max(max(readmatrix(app. ↙
bathymetryinputdata)));
1846                        end
1847                    end
1848                    %Enable the 'Land Color' dropdown list in 'Basemap' section
1849                    app.CoastlinecolourDropDown_2.Enable = "on";
1850                end
1851
1852                if isprop(app,'bathymetrydata')
1853                    [app.bathyRow,app.bathyCol] = size(app.bathymetrydata);
1854                end
1855
1856            end
1857
1858            % Callback function
1859            function ImportinfoButtonPushed(app, event)
1860
1861            end
1862
1863            % Value changed function: FileTextArea
1864            function FileTextAreaValueChanged(app, event)
1865                if contains(app.FileTextArea.Value, "hmax") || contains(app. ↙
FileTextArea.Value, "eta")
1866                    app.WaveHeightContoursButton.Enable = "on";
1867                end
1868
1869            end
1870
1871            % Callback function
```

```
1872         function ButtonGroup_2SelectionChanged(app, event)
1873
1874         end
1875
1876         % Callback function
1877         function MinimumEditFieldValueChanged(app, event)
1878             app.MinimumEditField.FontColor = 'k';
1879
1880         end
1881
1882         % Callback function
1883         function MaximumEditFieldValueChanged(app, event)
1884             app.MaximumEditField.FontColor = 'k';
1885         end
1886
1887         % Callback function
1888         function IntervalEditField_5ValueChanged(app, event)
1889             app.IntervalEditField_5.FontColor = 'k';
1890             app.IntervalEditField_3.Value = app.IntervalEditField_5.Value; %↙
Copy the values to the contour label interval textbox
1891             drawnow
1892
1893         end
1894
1895         % Value changed function: CoastlinecolourDropDown_8
1896         function CoastlinecolourDropDown_6ValueChanged(app, event)
1897             value = app.CoastlinecolourDropDown_8.Value;
1898             %List of values based on the cbrewer2 colormap
1899             validValues = {'blue', 'blue - green', 'blue - purple', 'green -↙
blue', ...
1900                 'greens', 'grays', 'oranges', 'orange - red', ...
1901                 'purple - blue', 'purple - blue - green', 'purple - red', ...
1902                 'purples', 'red - purple', 'reds', 'yellow - green', ...
1903                 'yellow - green - blue', 'yellow - orange - brown', ...
1904                 'yellow - orange - red', 'brown - teal', 'pink - light green',↙
...
1905                 'purple - green', 'purple - orange', 'red - blue', ...
1906                 'red - gray', 'red - yellow - blue', 'red - yellow - green',↙
...
1907                 'spectral', 'accent', 'dark 2', 'paired', 'pastel 1', ...
1908                 'pastel 2', 'set 1', 'set 2', 'set 3'};
1909
1910             % If the selected value in the dropdown list is part of the↙
cbrewer2 colormap, enable the color interpolation textbox
1911             if ismember(value, validValues)
1912                 app.DivisionEditField.Enable = "on";
1913                 app.DivisionEditFieldLabel.Enable = "on";
1914             else
1915                 app.DivisionEditField.Enable= "off";
1916                 app.DivisionEditFieldLabel.Enable = "off";
1917             end
1918
1919         end
1920
1921         % Callback function
1922         function Button_8Pushed(app, event)
1923             try
1924             [filename3,path3] = uigetfile('*.txt;*.shp');
```

```
1925            fullname = fullfile(path3,filename3);
1926
1927            [~, name, ext] = fileparts(filename3);
1928            if ~(ismcc || isdeployed)
1929                addpath(genpath(string(path3)));
1930            end
1931
1932            app.GaugeFname = name;
1933            app.GAUGEFILE = filename3;
1934            app.FileEditField.Value = filename3;
1935            app.FileEditField.FontColor = 'k';
1936            app.FileEditField.BackgroundColor = 'w';
1937
1938
1939            if strcmp(ext, '.shp')
1940                S = shaperead(fullname);
1941                latitudes= [S.Y]';
1942                longitudes= [S.X]';
1943                list = [round(latitudes, 4), round(longitudes, 4)];
1944                list = unique(list, 'rows');
1945                app.latGauge = list(:, 1);
1946                app.longGauge =  list(:, 2);
1947            elseif strcmp(ext, '.txt')
1948                fileID = fopen(fullname, 'r');
1949                dataArray = textscan(fileID, '%f %f', 'Delimiter',↙
'whitespace'); %Read text file and ensure that it is tab delimited
1950                fclose(fileID);
1951                app.STATION_FILE = string(filename3);
1952                app.latGauge  = dataArray{1};
1953                app.longGauge = dataArray{2};
1954            end
1955            end
1956
1957
1958
1959        end
1960
1961        % Value changed function: CoastlinecolourDropDown_2
1962        function CoastlinecolourDropDown_2ValueChanged(app, event)
1963
1964
1965        end
1966
1967        % Callback function
1968        function ButtonGroup_6SelectionChanged(app, event)
1969
1970        end
1971
1972        % Button pushed function: Button_2
1973        function Button_2Pushed(app, event)
1974            % Set up the directory path for saving the files
1975            workingfolder = uigetdir;
1976            Dir1 = fullfile(string(workingfolder), 'OUTPUT_FILES');
1977            app.WORKFOLDER = Dir1;
1978
1979            %Create the 'OUTPUT_FILES/Figures' folder in the working directory
1980            FigureFolder = fullfile(Dir1, 'Figures');
1981            app.WORKFOLDER0 = FigureFolder;
```

```
1982            app.OutputDirectoryEditField.Value = fullfile(deblank ↙
(FigureFolder));
1983            app.OutputDirectoryEditField.FontColor = 'k';
1984
1985            %Create the folders
1986            if ~exist(Dir1, 'dir')
1987                mkdir(Dir1);
1988            end
1989            if ~exist(FigureFolder, 'dir')
1990                mkdir(FigureFolder);
1991            end
1992        end
1993
1994        % Callback function
1995        function VirtualGaugesButtonValueChanged(app, event)
1996            %Display the gauges tab
1997            app.TabGroup2.SelectedTab = app.GaugesTab;
1998
1999
2000        end
2001
2002        % Callback function
2003        function BathymetryContoursButtonValueChanged(app, event)
2004
2005        end
2006
2007        % Callback function
2008        function WaveHeightContoursButtonValueChanged(app, event)
2009
2010
2011        end
2012
2013        % Value changed function: DepthFileEditField
2014        function DepthFileEditFieldValueChanged(app, event)
2015            value = app.DepthFileEditField.Value;
2016            if value == app.bathymetryname
2017                app.BathymetryContoursButton.Enable = "on";
2018            end
2019        end
2020
2021        % Callback function
2022        function LineintervalEditField_2ValueChanged(app, event)
2023            app.wh_interval.Value = app.LineintervalEditField_2.Value;
2024            drawnow
2025            app.LineintervalEditField_2.FontColor = 'k';
2026        end
2027
2028        % Callback function
2029        function MaximumdepthEditField_2ValueChanged(app, event)
2030
2031
2032            app.MaximumdepthEditField_2.FontColor = 'k';
2033        end
2034
2035        % Button pushed function: PLOTButton
2036        function PLOTButtonPushed(app, event)
2037            close all
2038            %------------------------
```

```
2039            %------Marker Style------
2040            %-----------------------
2041            GaugeMarkerOptions = containers.Map({'none', 'o', '+', '*', '.',↙
'x', '-', '|', '^', 'v', '>', '<', 'diamond', 'hexagram', 'pentagram', 'square'},↙
...
2042                {'none', 'o', '+', '*', '.', 'x', '-', '|', '^', 'v', '>',↙
'<', 'diamond', 'hexagram', 'pentagram', 'square'});
2043            GaugeMarker2 = GaugeMarkerOptions(app.MarkerDropDown.Value);
2044
2045
2046            %----------------------------
2047            %------Line Style at y=0------
2048            %----------------------------
2049            YLineStyleOptions = containers.Map({'None','Solid', 'Dashed',↙
'Dotted', 'Dash-dotted'}, {'off','-', '--', ':', '-.'});
2050            YLineStyle = YLineStyleOptions(app.Lineaty0DropDown.Value);
2051
2052            %----------------------
2053            %------Line Style------
2054            %----------------------
2055            GLineStyleOptions = containers.Map({'Solid', 'Dashed', 'Dotted',↙
'Dash-dotted'}, {'-', '--', ':', '-.'});
2056            GLineStyle = GLineStyleOptions(app.GaugeLineStyle.Value);
2057
2058            %-----------------------------------------
2059            %------Line Color for one data per plot------
2060            %-----------------------------------------
2061            colors = {'Black', 'Dark gray', 'Medium gray', 'Light gray',↙
'Red', 'Green', 'Blue', 'Yellow', 'Cyan', 'Magenta', 'White'};
2062            colorCodes = {'k', '[0.4 0.4 0.4]', '[0.7 0.7 0.7]', '[0.8 0.8↙
0.8]', 'r', 'g', 'b', 'y', 'c', 'm', 'w'};
2063            index = find(strcmp(colors, app.GaugeLineColor.Value));
2064            GLineColor0 = colorCodes{index};
2065
2066            %-------------------------------------------------
2067            %------Line Color for multiple data in one graph------
2068            %-------------------------------------------------
2069            LineColorOptions = containers.Map({'autumn', 'bone', 'colorcube',↙
'cool', 'copper', 'flag', ...
2070                'gray', 'hot', 'hsv', 'jet', 'parula', 'pink', 'prism',↙
'spring', 'summer', ...
2071                'turbo', 'winter', 'lines'}, ...
2072                {'autumn', 'bone', 'colorcube', 'cool', 'copper', 'flag',↙
'gray', 'hot', ...
2073                'hsv', 'jet', 'parula', 'pink', 'prism', 'spring', 'summer',↙
'turbo', 'winter', 'lines'});
2074
2075            GLineColor = LineColorOptions(app.GaugeLineColor_multiple.Value);
2076
2077            %--------------------------
2078            %------Legend Location------
2079            %--------------------------
2080            LegendLocationOptions = containers.Map({'North', 'South', 'East',↙
'West', 'Northeast', ...
2081                'Northwest', 'Southeast', 'Southwest', 'Northoutside',↙
'Southoutside', 'Eastoutside', ...
2082                'Westoutside', 'Northeastoutside', 'Northwestoutside',↙
'Southeastoutside', 'Southwestoutside', ...
```

```
2083                 'Best', 'Bestoutside', 'None'}, ...
2084                 {'north', 'south', 'east', 'west', 'northeast', 'northwest',↙
'southeast', 'southwest', ...
2085                 'northoutside', 'southoutside', 'eastoutside', 'westoutside',↙
'northeastoutside', ...
2086                 'northwestoutside', 'southeastoutside', 'southwestoutside',↙
'best', 'bestoutside', 'none'});
2087             legloc = LegendLocationOptions(app.LocationDropDown.Value);
2088             app.LegendLocation = legloc;
2089
2090             %--------------------------------------
2091             %------SAVE MAP: Output Directory------
2092             %--------------------------------------
2093             %Check if the output directory has been manually set by the user
2094             if  app.OutputDirectoryEditField_2.Value == string(app.↙
WORKFOLDER5)
2095                 inputtxt_directory3 =  fullfile(app.WORKFOLDER2);
2096                 Dir1 = fullfile(app.WORKFOLDER5);
2097                 app.OutputDirectoryEditField_2.FontColor = 'k';
2098                 if ~exist(inputtxt_directory3, 'dir')
2099                     mkdir(inputtxt_directory3);
2100                 end
2101                 if ~exist(Dir1, 'dir')
2102                     mkdir(Dir1);
2103                 end
2104             else
2105                 % Set the default directory to the Desktop
2106                 if ismac
2107                     %macOS
2108                     defaultDir = fullfile(getenv('HOME'), 'Desktop');
2109                 elseif ispc
2110                     %Windows
2111                     defaultDir = fullfile(getenv('USERPROFILE'), 'Desktop');
2112                 else
2113                     %Others
2114                     defaultDir = pwd;
2115                 end
2116
2117                 %Create 'OUTPUT_FILES' folder and 'Figures' subfolder
2118                 inputtxt_directory3 = fullfile(defaultDir, 'OUTPUT_FILES');
2119 %                 Dir1 = fullfile(defaultDir, 'OUTPUT_FILES/Figures');
2120                 Dir1 = fullfile(inputtxt_directory3,'Figures');
2121                 if ~exist(inputtxt_directory3, 'dir')
2122                     mkdir(inputtxt_directory3);
2123                 end
2124                 if ~exist(Dir1, 'dir')
2125                     mkdir(Dir1);
2126                 end
2127
2128                 %Display the path directory of the 'Figures' subfolder in the↙
textbox
2129                 app.OutputDirectoryEditField_2.Value = deblank(string(Dir1));
2130                 app.OutputDirectoryEditField_2.FontColor = 'k';
2131             end
2132
2133
2134             %----------------------
2135             %------LOG REPORT------
```

```
2136            %----------------------
2137            %Create 'Log_Files' subfolder in 'OUTPUT_FILES'
2138            LogFolder = fullfile(inputtxt_directory3, 'Log_Files');
2139            if ~exist(LogFolder, 'dir')
2140                mkdir(LogFolder);
2141            end
2142
2143            %Create the text file
2144            fileId = fopen(fullfile(LogFolder, 'Log_Report_GaugeRecords.↙
txt'),'w'); %if same file exists, overwrite the file
2145            fileList = app.FileTextArea_2.Value;
2146
2147            %Template for the default section of the log report (includes↙
information from the 'Input Data' section)
2148            fprintf(fileId, '%s\n','******************** LOG REPORT↙
******************** ');
2149            fprintf(fileId, '%-30s%-s\n', 'Type:', 'Gauge Records');
2150            timestamp = datestr(now, 'yyyy-mm-dd HH:MM:SS'); % Get the current↙
timestamp
2151            fprintf(fileId, '%-30s%-s\n','Timestamp:', timestamp); %Add↙
timestamp in the header
2152            fprintf(fileId, '%↙
s\n','________________________________________________');
2153            fprintf(fileId,'%-s\n\n','');
2154            fprintf(fileId,'%-30s%-s\n','X axis:',app.XAxisEditField.Value);
2155            fprintf(fileId,'%-30s%-s\n','Y axis:',app.YAxisEditField.Value);
2156
2157
2158            %------LOG REPORT--------------------
2159            %------Input Data: Import Files------
2160            %------------------------------------
2161            %If no file is uploaded
2162            if any(app.FileTextArea_2.Value == "0" | app.FileTextArea_2.Value↙
== "00") || isempty(fileList)
2163                fprintf(fileId, '%-30s%-s\n', 'IMPORT FILE/S:', 'Empty. No↙
data plotted');
2164            end
2165
2166            %Check if the name of each uploaded file starts with 'sta_'
2167            for i = 1:numel(fileList)
2168                if any(contains(fileList, 'sta_xxxx')) || ~any(contains↙
(fileList(i), 'sta_'))
2169                    fprintf(fileId, '%-30s%-s\n', 'IMPORT FILE/S:', 'Invalid↙
filename format. Use filenames with "sta_"');
2170                    fprintf(fileId, '%-30s%-s\n', '', string(fileList(i)));
2171                end
2172            end
2173
2174            %List all the files loaded
2175            fprintf(fileId, '%-30s%-s\n', 'Loaded Files:', [num2str(length↙
(app.GaugeDirectorylist)),' files']);
2176            fprintf(fileId, '%↙
s\n\n','_____________________________________________________');
2177
2178            %------LOG REPORT--------------------------
2179            %------General Layout: Boundary Limits------
2180            %-------------------------------------------
2181            %Check the X limits and interval of the plot if 'Auto Set' is not↙
```

```
selected for the limits
2182            if ~app.CheckBox_2.Value
2183                if app.xMinLimit.Value > app.xMaxLimit.Value
2184                    fprintf(fileId, '%-30s%-s\n', 'X Limit:','Max value must↙
exceed min value');
2185                    fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved: Max and min↙
values are swapped');
2186                end
2187                if app.xLimInterval.Value > app.xMaxLimit.Value - app.↙
xMinLimit.Value
2188                    fprintf(fileId, '%-30s%-s\n', 'X Limit Interval:','Value↙
must be less than the max-min difference');
2189                    fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved: Value↙
adjusted to half the max-min difference');
2190                end
2191            end
2192
2193            %Check the Y limits and interval of the plot if 'Auto Set' is not↙
selected for the limits
2194            if ~app.CheckBox.Value
2195                if app.yMinLimit.Value > app.yMaxLimit.Value
2196                    fprintf(fileId, '%-30s%-s\n', 'Y Limit:','Max value must↙
exceed min value');
2197                    fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved: Max and min↙
values are swapped');
2198                end
2199                if  app.yLimInterval.Value > app.yMinLimit.Value - app.↙
yMaxLimit.Value
2200                    fprintf(fileId, '%-30s%-s\n', 'Y Limit Interval:','Value↙
must be less than the max-min difference');
2201                    fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved: Value↙
adjusted to half the max-min difference');
2202                end
2203            end
2204
2205            %% PROCESS THE INPUT FILES AND VALUES, THEN GENERATE THE FIGURES
2206            %--------------------------------
2207            %------READ THE INPUT FILES------
2208            %--------------------------------
2209            clear xData yData yData2
2210            %Check if any file is loaded
2211            if app.FileTextArea_2.Value ~=  string(app.GaugeNames)
2212                app.FileTextArea_2.Value = "NO FILE";
2213            else
2214                %Read the data
2215                FileLength = length(app.GaugeDirectorylist);
2216                plotHandles = [];
2217                legendEntries = {};
2218
2219                for i = 1:FileLength
2220                    g = readmatrix(app.GaugeDirectorylist(i)); %Read all↙
matrices
2221
2222                    %Calculate the magnitude of the Z-component of velocity↙
using the U and V velocity values in columns 3 and 4.
2223                    Zvelocity = sqrt(g(:,3).^2 + g(:,4).^2);
2224                    %Include the Z velocity as the fifth column
2225                    g = [g, Zvelocity];
```

```
2226                        %Extracting the station numbers from the filename
2227                        GN = erase(app.GaugeNames(i),"sta_");  %remove 'sta_' from the name
2228                        GN2 = strip(GN,"left",'0'); %remove the trailing zeroes on the left side
2229
2230                        if contains(GN2,'_')
2231                            GaugeNumber{i} = regexprep(GN2,'_',' '); %Replace "_" to space
2232                        else
2233                            GaugeNumber{i} = GN2;
2234                        end
2235
2236                        %Choose the column from which to extract data points for the X plot values
2237                        app.XCOLGAUGE = str2double(app.XaxisUseDataDropDown.Value);
2238                        app.YCOLGAUGE = str2double(app.YaxisUseDataDropDown.Value);
2239
2240                        %Convert data loaded for X axis
2241                        if app.CheckBox_3.Value
2242                            g(:,app.XCOLGAUGE) =  g(:,app.XCOLGAUGE) * (app.numeratorX.Value/app.denomeratorX.Value);
2243                        end
2244
2245                       %If '3,4,5' or '5' option is selected, calculate Z velocity vector
2246                        if strcmp(app.YaxisUseDataDropDown.Value, '3,4,5') ||strcmp(app.YaxisUseDataDropDown.Value, '5')
2247                        %Calculate the magnitude of the Z-component of velocity using the U and V velocity values in columns 3 and 4.
2248                        Zvelocity = sqrt(g(:,3).^2 + g(:,4).^2);
2249                        %Include the Z velocity as the fifth column
2250                        g = [g, Zvelocity];
2251                        end
2252
2253                        %Convert data loaded for Y axis
2254                        if app.CheckBox_4.Value %Convert data loaded for Y axis
2255                            if ~strcmp(app.YaxisUseDataDropDown.Value,'3,4,5') %If only one column of data will be used
2256                                g(:,app.YCOLGAUGE) =  g(:,app.YCOLGAUGE) * (app.numeratorY.Value / app.denomeratorY.Value);
2257                            else
2258                                %Convert the values
2259                                g(:, 3) =  g(:, 3) * (app.numeratorY.Value / app.denomeratorY.Value);
2260                                g(:, 4) =  g(:, 4) * (app.numeratorY.Value / app.denomeratorY.Value);
2261                                g(:, 5) =  g(:, 5) * (app.numeratorY.Value / app.denomeratorY.Value);
2262                            end
2263                        end
2264
2265                        %Extract the required Y data based on dropdown selection
2266                        if ~strcmp(app.YaxisUseDataDropDown.Value, '3,4,5')
```

```
2267                            yData = g(:,str2double(app.YaxisUseDataDropDown.↙
Value));
2268                        else
2269                            yData = g(:, 3);
2270                            y2Data = g(:, 4);
2271                            y3Data = g(:, 5);
2272                        end
2273
2274                        %Sort data list to avoid connecting lines between the↙
first and last points
2275                        xData = double(g(:, app.XCOLGAUGE));
2276                        [xDataArranged, sortIndex] = sort(xData);
2277                        yDataArranged= yData(sortIndex);
2278                        if strcmp(app.YaxisUseDataDropDown.Value, '3,4,5') %If↙
columns 3, 4, 5 are to be plotted together
2279                            y2DataArranged= y2Data(sortIndex);
2280                            y3DataArranged= y3Data(sortIndex);
2281                        end
2282
2283                        %---------------------
2284                        %------MAIN PLOT------
2285                        %---------------------
2286                        %Select the line color of the plots
2287                        if app.PlotalldatainonegraphButton.Value
2288                            a = str2func(GLineColor);
2289                            rndm = a(FileLength); %divide the colormap based↙
on the number of uploaded dataset
2290
2291                            if app.FlipCheckBox_3.Value % Flip colors
2292                                app.colorcombi = rndm(FileLength-i+1,:);
2293                            else
2294                                app.colorcombi = rndm(i,:);
2295                            end
2296                        elseif app.PlotseparatelyButton_4.Value
2297                            app.colorcombi = GLineColor0; %For single line↙
plot
2298                        end
2299
2300                        %--------------------------------------
2301                        %------PLOT ALL DATA IN ONE GRAPH------
2302                        %--------------------------------------
2303                            if app.PlotalldatainonegraphButton.Value %Plot all↙
sta_ data into one figure
2304                                figure(1);
2305                                set(gcf, 'Color', 'w');
2306                                hold on;
2307
2308                                if strcmp(app.YaxisUseDataDropDown.Value,↙
'3,4,5')
2309                                    %Main plotting loop
2310                                    %U vector
2311                                    ph1 = plot(xDataArranged, yDataArranged,↙
'LineWidth', app.DataLineWidth.Value,'LineStyle', '--', 'Marker', GaugeMarker2, ...
2312                                        'MarkerEdgeColor', app.colorcombi,↙
'MarkerFaceColor', app.colorcombi, 'Color', [0.596, 0.404, 0.553]);
2313                                    hold on
2314                                    %V vector
2315                                    ph2 = plot(xDataArranged, y2DataArranged,↙
```

```
'LineWidth', app.DataLineWidth.Value, 'LineStyle','--','Marker',↙
GaugeMarker2,'MarkerEdgeColor',app.colorcombi,'MarkerFaceColor',app.↙
colorcombi,'Color', [0.4, 0.596, 0.447]);
2316                                    hold on
2317                                    %Z vector
2318                                    ph3 = plot(xDataArranged, y3DataArranged,↙
'LineWidth', app.DataLineWidth.Value, 'LineStyle', GLineStyle,'Marker',↙
GaugeMarker2,'MarkerEdgeColor',app.colorcombi,'MarkerFaceColor',app.↙
colorcombi,'Color','k');
2319                                    hold off
2320
2321
2322                                    plotHandles(end+1) = ph1;
2323                                    plotHandles(end+1) = ph2;
2324                                    plotHandles(end+1) = ph3;
2325
2326                                    legendEntries{end+1} = sprintf('%s %s U↙
vector', string(app.legendfirsttext.Value), string(GaugeNumber(min(i, length↙
(GaugeNumber)))));
2327                                    legendEntries{end+1} = sprintf('%s %s V↙
vector', string(app.legendfirsttext.Value), string(GaugeNumber(min(i, length↙
(GaugeNumber)))));
2328                                    legendEntries{end+1} = sprintf('%s %s Z↙
vector', string(app.legendfirsttext.Value), string(GaugeNumber(min(i, length↙
(GaugeNumber)))));
2329
2330                                end %Plot data from only one column
2331                                    ph1 = plot(xDataArranged, yDataArranged,↙
'LineWidth', app.DataLineWidth.Value,'LineStyle', GLineStyle, 'Marker',↙
GaugeMarker2, ...
2332                                        'MarkerEdgeColor', app.colorcombi,↙
'MarkerFaceColor', app.colorcombi, 'Color', app.colorcombi);
2333                                    plotHandles(end+1) = ph1;  % Store first↙
plot handle
2334                                    legendEntries{end+1} = sprintf('%s %s',↙
string(app.legendfirsttext.Value), string(GaugeNumber(min(i, length↙
(GaugeNumber)))));
2335                                end
2336
2337                                % Update legend outside the loop with all↙
entries
2338                                legend(plotHandles, legendEntries, 'Location',↙
legloc, 'FontSize', app.LegendSize.Value);
2339                                legend box off;
2340                                hold off;
2341
2342                                %---------------------------------------
2343                                %------GENERAL LAYOUT: Figure Size-------
2344                                %---------------------------------------
2345                                if ~app.AutoSetCheckBox.Value
2346                                    %Manually set the figure size
2347                                    figureHandle = ancestor(ph1, 'figure');
2348                                    set(figureHandle, 'Units', 'Inches',↙
'Position', [0, 0, app.Width.Value, app.Height.Value], 'PaperUnits', 'Inches',↙
'PaperSize', [app.Width.Value, app.Height.Value]);
2349                                    set(gcf, 'Units', 'Inches', 'Position',↙
[0, 0, app.Width.Value, app.Height.Value], 'PaperUnits', 'Inches', 'PaperSize',↙
[app.Width.Value, app.Height.Value]);
```

```
2350                                    end
2351
2352                                end
2353
2354                                %---------------------------
2355                                %------PLOT SEPARATELY------
2356                                %---------------------------
2357                                if app.PlotseparatelyButton_4.Value %Plot each↙
sta_ data in separate figures
2358
2359                                    % Open a new figure for each file
2360                                    figure(i);
2361                                    clf %clean figure so legend won't repeat
2362                                    set(gcf, 'Color', 'w'); % Set figure↙
background to white
2363
2364                                    % Reset legend values for each figure
2365                                    plotHandles = [];
2366                                    legendEntries = {};
2367
2368                                    hold on
2369                                    if strcmp(app.YaxisUseDataDropDown.Value,↙
'3,4,5')
2370                                        %Main plotting loop
2371                                        %U vector
2372                                        ph1 = plot(xDataArranged, yDataArranged,↙
'LineWidth', app.DataLineWidth.Value,'LineStyle', '--', 'Marker', GaugeMarker2, ...
2373                                            'MarkerEdgeColor', app.colorcombi,↙
'MarkerFaceColor', app.colorcombi, 'Color', [0.596, 0.404, 0.553]);
2374                                        hold on
2375                                        %V vector
2376                                        ph2 = plot(xDataArranged, y2DataArranged,↙
'LineWidth', app.DataLineWidth.Value, 'LineStyle', '--','Marker',↙
GaugeMarker2,'MarkerEdgeColor',app.colorcombi,'MarkerFaceColor',app.↙
colorcombi,'Color',[0.4, 0.596, 0.447]);
2377                                        disp(ph2);
2378                                        hold on
2379                                        %Z vector
2380                                        ph3 = plot(xDataArranged, y3DataArranged,↙
'LineWidth', app.DataLineWidth.Value, 'LineStyle', GLineStyle,'Marker',↙
GaugeMarker2,'MarkerEdgeColor',app.colorcombi,'MarkerFaceColor',app.↙
colorcombi,'Color','k');
2381                                        hold off
2382
2383                                        plotHandles(end+1) = ph1;
2384                                        plotHandles(end+1) = ph2;
2385                                        plotHandles(end+1) = ph3;
2386
2387                                        %Add each legend entry
2388                                        legendEntries{end+1} = 'U vector';
2389                                        legendEntries{end+1} = 'V vector';
2390                                        legendEntries{end+1} = 'Z vector';
2391
2392                                        legend(plotHandles, legendEntries,↙
'Location', legloc, 'FontSize', app.LegendSize.Value);
2393                                        legend box off;
2394
2395                                    else %Plot data from only one column
```

```
2396                                    ph1 = plot(xDataArranged, yDataArranged,↙
'LineWidth', app.DataLineWidth.Value,'LineStyle', GLineStyle, 'Marker',↙
GaugeMarker2, ...
2397                                        'MarkerEdgeColor', app.colorcombi,↙
'MarkerFaceColor', app.colorcombi, 'Color', app.colorcombi);
2398                                    end
2399
2400                                    titleText = sprintf('%s %s', string(app.↙
legendfirsttext.Value), string(GaugeNumber(min(i, length(GaugeNumber)))));
2401                                    title(titleText);
2402
2403                                    %Figure Size
2404                                    if ~app.AutoSetCheckBox.Value
2405                                        figureHandle = ancestor(ph1, 'figure'); %↙
Get the figure handle containing the axes
2406                                        set(figureHandle, 'Units', 'Inches',↙
'Position', [0, 0, app.Width.Value, app.Height.Value], 'PaperUnits', 'Inches',↙
'PaperSize', [app.Width.Value, app.Height.Value]);
2407                                        set(gcf, 'Units', 'Inches', 'Position',↙
[0, 0, app.Width.Value, app.Height.Value], 'PaperUnits', 'Inches', 'PaperSize',↙
[app.Width.Value, app.Height.Value]);
2408                                    end
2409                                end
2410
2411
2412                                %-------------------------------------------
2413                                %------GENERAL LAYOUT: Boundary Limit-------
2414                                %-------------------------------------------
2415                                %Set X axis limit
2416                                if app.CheckBox_2.Value
2417                                    xlim("auto")
2418                                    xticks('auto')
2419                                else %Using the input values
2420                                    %Swap the minimum and maximum input values of↙
X-axis limits when needed
2421                                    if app.xMinLimit.Value > app.xMaxLimit.Value
2422                                        temp3 = app.xMinLimit.Value;
2423                                        app.xMinLimit.Value = app.xMaxLimit.Value;
2424                                        app.xMaxLimit.Value = temp3;
2425                                    end
2426
2427                                    %Check the interval value if it is within the↙
limits
2428                                    if app.xLimInterval.Value > (app.xMaxLimit.↙
Value-app.xMinLimit.Value)
2429                                        app.xLimInterval.Value = (app.xMaxLimit.↙
Value-app.xMinLimit.Value)/2;
2430                                    end
2431
2432                                    xlim([app.xMinLimit.Value app.xMaxLimit.↙
Value])
2433                                    xticksLOC = app.xMinLimit.Value:app.↙
xLimInterval.Value:app.xMaxLimit.Value;
2434                                    xticks(xticksLOC)
2435                                    tickXname = string(xticksLOC);
2436                                    xticklabels(tickXname)
2437                                end
2438
```

```
2439                                %Set Y axis limit
2440                                if app.CheckBox.Value
2441                                    ylim("auto")
2442                                    yticks('auto')
2443                                else %Using the input values
2444                                    % Swap the minimum and maximum input values of
Y-axis limits when needed
2445                                    if app.yMinLimit.Value > app.yMaxLimit.Value
2446                                        temp4 = app.yMinLimit.Value;
2447                                        app.yMinLimit.Value = app.yMaxLimit.Value;
2448                                        app.yMaxLimit.Value = temp4;
2449                                    end
2450
2451                                    %Check the interval value if it is within the
limits
2452                                    if app.yLimInterval.Value > (app.yMaxLimit.
Value-app.yMinLimit.Value) %Check the interval value if it is within the limits
2453                                        app.yLimInterval.Value = (app.yMaxLimit.
Value-app.yMinLimit.Value)/2;
2454                                    end
2455
2456                                    ylim([app.yMinLimit.Value app.yMaxLimit.
Value]);
2457                                    yticksLOC = [app.yMinLimit.Value:app.
yLimInterval.Value:app.yMaxLimit.Value];
2458                                    yticks(yticksLOC)
2459                                    tickYname = string(yticksLOC);
2460                                    yticklabels(tickYname)
2461                                    ytickformat('%.1f')
2462                                end
2463
2464
2465                            %-------------------------------------------
2466                            %------GENERAL LAYOUT: Background Grid-------
2467                            %-------------------------------------------
2468                            ax = gca;
2469                            ax.FontSize = app.GridLabelSize.Value;
2470                            ax.LineWidth = app.LineYThickness_2.Value;
2471                            if strcmp(app.GridStyle.Value, 'none')
2472                                grid(ax, 'off');
2473                            elseif strcmp(app.GridStyle.Value, 'X axis only')
2474                                ax.XGrid = 'on';
2475                                ax.YGrid = 'off';
2476                                ax.XMinorGrid = 'off';
2477                                ax.YMinorGrid = 'off';
2478                            elseif strcmp(app.GridStyle.Value, 'Y axis only')
2479                                ax.XGrid = 'off';
2480                                ax.YGrid = 'on';
2481                                ax.XMinorGrid = 'off';
2482                                ax.YMinorGrid = 'off';
2483                            elseif strcmp(app.GridStyle.Value, 'Both - major
lines')
2484                                grid(ax, 'on');
2485                                ax.XMinorGrid = 'off';
2486                                ax.YMinorGrid = 'off';
2487                            elseif strcmp(app.GridStyle.Value, 'Both - with minor
lines')
2488                                grid(ax, 'on');
```

```
2489                            ax.XMinorGrid = 'on';
2490                            ax.YMinorGrid = 'on';
2491                        end
2492
2493                        %Grid line at y = 0 for all data in one graph
2494                        if ~strcmp(app.Lineaty0DropDown.Value,"None")
2495                            yhorzline = yline(0,YLineStyle,'LineWidth',app.↙
LineYThickness.Value);
2496                            yhorzline.Annotation.LegendInformation.↙
IconDisplayStyle = 'off';
2497                        end
2498
2499
2500                        %----------------------------------
2501                        %------GENERAL LAYOUT: Ticks-------
2502                        %----------------------------------
2503                        %Labels of Axes and Ticks
2504                        xlabel(app.XAxisEditField.Value,'FontSize',app.↙
GridLabelSize.Value);
2505                        ylabel(app.YAxisEditField.Value,'FontSize',app.↙
GridLabelSize.Value);
2506
2507
2508                        %---------------------------------------
2509                        %------SAVE MAP: Plot Separately-------
2510                        %---------------------------------------
2511                        if app.PlotseparatelyButton_4.Value
2512                            %Save the maps with filenames matching the loaded↙
file
2513                            PlotsFolder = fullfile(Dir1);
2514                            OF = fullfile(PlotsFolder, app.GaugeNames(i));
2515
2516                            %Options for saving format
2517                            %PNG
2518                            if app.pngCheckBox_2.Value
2519                                outputfile = OF + ".png";
2520                                exportgraphics(ax,outputfile,'Resolution',300)
2521                            end
2522
2523                            %JPG
2524                            if app.jpgCheckBox_2.Value
2525                                outputfile = OF + ".jpg";
2526                                exportgraphics(ax,outputfile,'Resolution',300)
2527                            end
2528
2529                            %PDF
2530                            if app.pdfCheckBox_2.Value
2531                                outputfile = OF + ".pdf";
2532                                exportgraphics(ax,↙
outputfile,'ContentType','vector','Resolution',300)
2533                            end
2534
2535                            %EPS
2536                            if app.epsCheckBox.Value %save as .eps
2537                                outputfile = OF + ".eps";
2538                                saveas(ax,outputfile,'epsc')
2539                            end
2540
```

```
2541                            %TXT for saving the raw 'sta_' files
2542                            if app.txtCheckBox.Value
2543                                outputfile = OF + ".txt";
2544                                writematrix(g, outputfile, 'Delimiter', '\t');
2545                                dlmwrite(outputfile,g,'delimiter', '\t');
2546                            end
2547                        end
2548                end
2549
2550                %Add legend in the plot
2551                if app.PlotseparatelyButton_4.Value
2552                    if strcmp(app.YaxisUseDataDropDown.Value, '3,4,5')
2553                        legend(plotHandles, legendEntries, 'Location', legloc,↙
'FontSize', app.LegendSize.Value);
2554                        legend box off;
2555                    end
2556                end
2557
2558                %-------------------------------------------------
2559                %------SAVE MAP: Plot all data in one graph-------
2560                %-------------------------------------------------
2561                if app.PlotalldatainonegraphButton.Value
2562                    if length(app.GaugeNames) == 1
2563                        OF = fullfile(Dir1, app.GaugeNames);
2564                    else
2565                        OF = fullfile(Dir1, 'GaugeRecord');
2566                    end
2567
2568                    if app.jpgCheckBox_2.Value
2569                        outputfile = OF + ".jpg";
2570                        exportgraphics(gcf,outputfile,'Resolution',300)
2571                    end
2572
2573                    if app.pngCheckBox_2.Value
2574                        outputfile = OF + ".png";
2575                        exportgraphics(gcf,outputfile,'Resolution',300)
2576                    end
2577                    if app.pdfCheckBox_2.Value %save as .pdf
2578                        outputfile = OF + ".pdf";
2579                        exportgraphics(ax,↙
outputfile,'ContentType','vector','Resolution',300)
2580                    end
2581                    if app.epsCheckBox.Value %save as .eps
2582                        outputfile = OF + ".eps";
2583                        saveas(gcf,outputfile,'epsc')
2584                        pause(2)
2585                        clf;
2586                    end
2587                    if app.txtCheckBox.Value %save as .txt
2588                        outputfile = OF + ".txt";
2589                        writematrix(g,outputfile);
2590                        dlmwrite(outputfile,g,'delimiter', '\t');
2591                    end
2592                end
2593            end
2594        end
2595
2596        % Callback function
```

```
2597         function BathymetryCheckBoxValueChanged(app, event)
2598
2599         end
2600
2601         % Callback function
2602         function WaveHeightCheckBoxValueChanged(app, event)
2603
2604
2605         end
2606
2607         % Value changed function: GaugesCheckBox
2608         function GaugesCheckBoxValueChanged(app, event)
2609             if app.GaugesCheckBox.Value
2610                 %Display and enable the parameter options for the gauges tab
2611                 app.TabGroup2.SelectedTab = app.GaugesTab;
2612                 for i = 1:length(app.GaugesTab.Children)
2613                         app.GaugesTab.Children(i).Enable = 'on';
2614                 end
2615                     %If 'Add Labels' is checked, enable the items below it
2616                     components = {app.GLspacing, app.GLspacingLabel, app.↙
SizeEditField_3, app.SizeEditField_3Label, app.CoastlinecolourDropDown_5, app.↙
CoastlinecolourDropDown_4, app.AlignmentLabel};
2617
2618                         for i = 1:length(components)
2619                              if  app.AddLabelsCheckBox_4.Value
2620                                  components{i}.Enable = 'on';
2621                              else
2622                                  components{i}.Enable = 'off';
2623                              end
2624                         end
2625             else
2626                 for i = 1:length(app.GaugesTab.Children)
2627                     if isprop(app.GaugesTab.Children(i), 'Enable')
2628                         app.GaugesTab.Children(i).Enable = 'off';
2629                     end
2630                 end
2631             end
2632         end
2633
2634         % Button pushed function: Button_16
2635         function Button_16Pushed(app, event)
2636              %Find files to import from a directory
2637             [files, path] = uigetfile('*.*','Select the files','MultiSelect',↙
'on');
2638             allfiles = string(path) + string(files);
2639             if ~(ismcc || isdeployed)
2640                 addpath(genpath(fullfile(string(path))));
2641             end
2642
2643             cd(fullfile(path))
2644
2645             %Sort the files by snapshot timing, and display the list of↙
filenames in the app's textbox
2646             app.GaugeDirectorylist = sort(allfiles);
2647             app.GaugeNames = sort(string(files));
2648             app.FileTextArea_2.Value = string(app.GaugeNames);
2649             app.FileTextArea_2.FontColor = 'k';
2650         end
```

```
2651
2652        % Value changed function: FileTextArea_2
2653        function FileTextArea_2ValueChanged(app, event)
2654
2655
2656        end
2657
2658        % Value changed function: CheckBox
2659        function CheckBoxValueChanged(app, event)
2660            if app.CheckBox.Value
2661                app.yMaxLimit.Enable = "off";
2662                app.yMinLimit.Enable = "off";
2663                app.yLimInterval.Enable = "off";
2664            else
2665                app.yMaxLimit.Enable = "on";
2666                app.yMinLimit.Enable = "on";
2667                app.yLimInterval.Enable = "on";
2668            end
2669        end
2670
2671        % Value changed function: CheckBox_2
2672        function CheckBox_2ValueChanged(app, event)
2673            if app.CheckBox_2.Value
2674                app.xMaxLimit.Enable = "off";
2675                app.xMinLimit.Enable = "off";
2676                app.xLimInterval.Enable = "off";
2677            else
2678                app.xMaxLimit.Enable = "on";
2679                app.xMinLimit.Enable = "on";
2680                app.xLimInterval.Enable = "on";
2681            end
2682        end
2683
2684        % Value changed function: yLimInterval
2685        function yLimIntervalValueChanged(app, event)
2686            app.yLimInterval.FontColor = 'k';
2687
2688
2689        end
2690
2691        % Value changed function: xLimInterval
2692        function xLimIntervalValueChanged(app, event)
2693
2694
2695        end
2696
2697        % Button pushed function: Button_18
2698        function Button_18Pushed(app, event)
2699            % Set up the directory path for saving the files
2700            workingfolder = uigetdir;
2701            Dir1 = fullfile(string(workingfolder), 'OUTPUT_FILES');
2702            app.WORKFOLDER2 = Dir1;
2703
2704            %Create the 'Figures' folder in the 'OUTPUT_FILES' working↙
directory
2705            FigureFolder = fullfile(Dir1, 'Figures');
2706            app.WORKFOLDER5 = FigureFolder;
2707            app.OutputDirectoryEditField_2.Value = FigureFolder;
```

```
2708            app.OutputDirectoryEditField_2.FontColor = 'k';
2709
2710            %Create the folders
2711            if ~exist(Dir1, 'dir')
2712                mkdir(Dir1);
2713            end
2714            if ~exist(FigureFolder, 'dir')
2715                mkdir(FigureFolder);
2716            end
2717
2718        end
2719
2720        % Value changed function: yMinLimit
2721        function yMinLimitValueChanged(app, event)
2722            app.yMinLimit.FontColor = 'k';
2723        end
2724
2725        % Value changed function: yMaxLimit
2726        function yMaxLimitValueChanged(app, event)
2727            app.yMaxLimit.FontColor = 'k';
2728        end
2729
2730        % Value changed function: xMinLimit
2731        function xMinLimitValueChanged(app, event)
2732            app.xMinLimit.FontColor = 'k';
2733
2734
2735        end
2736
2737        % Value changed function: xMaxLimit
2738        function xMaxLimitValueChanged(app, event)
2739
2740        end
2741
2742        % Value changed function: GridStyle
2743        function GridStyleValueChanged(app, event)
2744            value = app.GridStyle.Value;
2745            if strcmp(value,"None")
2746                app.LineYThickness_2.Enable = "off";
2747                app.LineYThickness_2.Editable = "off";
2748            else
2749                app.LineYThickness_2.Enable = "on";
2750                app.LineYThickness_2.Editable = "on";
2751            end
2752
2753        end
2754
2755        % Drop down opening function: GridStyle
2756        function GridStyleDropDownOpening(app, event)
2757
2758        end
2759
2760        % Value changed function: Lineaty0DropDown
2761        function Lineaty0DropDownValueChanged(app, event)
2762            value = app.Lineaty0DropDown.Value;
2763            if strcmp(value,"None")
2764                app.LineYThickness.Enable = "off";
2765            else
```

```
2766                 app.LineYThickness.Enable = "on";
2767             end
2768         end
2769 
2770         % Value changed function: LocationDropDown
2771         function LocationDropDownValueChanged(app, event)
2772             value = app.LocationDropDown.Value;
2773             if strcmp(value,"None")
2774                 app.LegendSize.Enable = "off";
2775             else
2776                 app.LegendSize.Enable = "on";
2777             end
2778 
2779         end
2780 
2781         % Callback function
2782         function XCOLValueChanged(app, event)
2783 
2784 
2785         end
2786 
2787         % Callback function
2788         function YCOLValueChanged(app, event)
2789 
2790         end
2791 
2792         % Value changed function: DataLineWidth
2793         function DataLineWidthValueChanged(app, event)
2794             app.DataLineWidth.FontColor = 'k';
2795 
2796         end
2797 
2798         % Value changed function: XAxisEditField
2799         function XAxisEditFieldValueChanged(app, event)
2800             app.XAxisEditField.FontColor = 'k';
2801 
2802         end
2803 
2804         % Value changed function: YAxisEditField
2805         function YAxisEditFieldValueChanged(app, event)
2806             app.YAxisEditField.FontColor = 'k';
2807 
2808         end
2809 
2810         % Value changed function: GridLabelSize
2811         function GridLabelSizeValueChanged(app, event)
2812             app.GridLabelSize.FontColor = 'k';
2813 
2814         end
2815 
2816         % Value changed function: LineYThickness
2817         function LineYThicknessValueChanged(app, event)
2818             app.LineYThickness.FontColor = 'k';
2819 
2820         end
2821 
2822         % Callback function
2823         function GridThicknessValueChanged(app, event)
```

```
2824            app.GridThickness.FontColor = 'k';
2825
2826        end
2827
2828        % Value changed function: LegendSize
2829        function LegendSizeValueChanged(app, event)
2830            app.LegendSize.FontColor = 'k';
2831
2832        end
2833
2834        % Value changed function: NorthEditField
2835        function NorthEditFieldValueChanged(app, event)
2836            app.NorthEditField.FontColor = 'k';
2837
2838
2839        end
2840
2841        % Value changed function: SouthEditField
2842        function SouthEditFieldValueChanged(app, event)
2843            app.SouthEditField.FontColor = 'k';
2844
2845        end
2846
2847        % Value changed function: EastEditField
2848        function EastEditFieldValueChanged(app, event)
2849            app.EastEditField.FontColor = 'k';
2850
2851        end
2852
2853        % Value changed function: WestEditField
2854        function WestEditFieldValueChanged(app, event)
2855            app.WestEditField.FontColor = 'k';
2856
2857        end
2858
2859        % Value changed function: toEditField
2860        function toEditFieldValueChanged(app, event)
2861            app.toEditField.FontColor = 'k';
2862
2863        end
2864
2865        % Value changed function: MinEditField
2866        function MinEditFieldValueChanged(app, event)
2867            app.MinEditField.FontColor = 'k';
2868
2869        end
2870
2871        % Value changed function: MaxEditField_2
2872        function MaxEditField_2ValueChanged(app, event)
2873            app.MaxEditField_2.FontColor = 'k';
2874
2875        end
2876
2877        % Value changed function: MinEditField_2
2878        function MinEditField_2ValueChanged(app, event)
2879            app.MinEditField_2.FontColor = 'k';
2880
2881        end
```

```
2882
2883        % Callback function
2884        function LabelSpacingEditFieldValueChanged(app, event)
2885            app.LabelSpacingEditField.FontColor = 'k';
2886
2887        end
2888
2889        % Callback function
2890        function IntervalEditField_3ValueChanged(app, event)
2891            app.IntervalEditField_3.FontColor = 'k';
2892
2893        end
2894
2895        % Callback function
2896        function WidthEditFieldValueChanged(app, event)
2897            app.WidthEditField.FontColor = 'k';
2898
2899        end
2900
2901        % Value changed function: ColorinterpolationEditField_2
2902        function ColorinterpolationEditField_2ValueChanged(app, event)
2903            app.ColorinterpolationEditField_2.FontColor = 'k';
2904
2905        end
2906
2907        % Value changed function: DivisionEditField
2908        function DivisionEditFieldValueChanged(app, event)
2909            app.DivisionEditField.FontColor = 'k';
2910
2911        end
2912
2913        % Callback function
2914        function MinimumdepthEditField_2ValueChanged(app, event)
2915            app.MinimumdepthEditField_2.FontColor = 'k';
2916
2917        end
2918
2919        % Callback function
2920        function ThicknessEditField_2ValueChanged(app, event)
2921            app.ThicknessEditField_2.FontColor = 'k';
2922
2923        end
2924
2925        % Callback function
2926        function wh_intervalValueChanged(app, event)
2927            app.wh_interval.FontColor = 'k';
2928        end
2929
2930        % Callback function
2931        function LabelSpacingEditField_2ValueChanged(app, event)
2932            app.LabelSpacingEditField_2.FontColor = 'k';
2933
2934        end
2935
2936        % Callback function
2937        function SizeEditFieldValueChanged(app, event)
2938            app.SizeEditField.FontColor = 'k';
2939
```

```
2940         end
2941
2942         % Callback function
2943         function SizeEditField_3ValueChanged(app, event)
2944             app.SizeEditField_3.FontColor = 'k';
2945
2946         end
2947
2948         % Value changed function: CheckBox_3
2949         function CheckBox_3ValueChanged(app, event)
2950             %If checked, activate the textboxes for X-axis data in the↙
'Convert Values' section of the 'Input Data'
2951             if app.CheckBox_3.Value
2952                 app.denomeratorX.Enable = "on";
2953                 app.numeratorX.Enable = "on";
2954             else
2955                 app.denomeratorX.Enable = "off";
2956                 app.numeratorX.Enable = "off";
2957             end
2958         end
2959
2960         % Value changed function: CheckBox_4
2961         function CheckBox_4ValueChanged(app, event)
2962             %If checked, activate the textboxes for Y-axis data in the↙
'Convert Values' section of the 'Input Data'
2963             if app.CheckBox_4.Value
2964                 app.denomeratorY.Enable = "on";
2965                 app.numeratorY.Enable = "on";
2966             else
2967                 app.denomeratorY.Enable = "off";
2968                 app.numeratorY.Enable = "off";
2969             end
2970         end
2971
2972         % Callback function
2973         function XaxisEditFieldValueChanging(app, event)
2974
2975             if event.Value
2976                 app.egconvertcolumn1datafromsecondstominutesLabel.Visible =↙
"off";
2977             end
2978         end
2979
2980         % Callback function
2981         function YaxisEditFieldValueChanging(app, event)
2982             if event.Value
2983                 app.egconvertcolumn2datafrommeterstocentimetersLabel.Visible =↙
"off";
2984             end
2985         end
2986
2987         % Value changed function: denomeratorY
2988         function denomeratorYValueChanged(app, event)
2989
2990         end
2991
2992         % Selection changed function: ButtonGroup_10
2993         function ButtonGroup_10SelectionChanged(app, event)
```

```
2994            %Activate/deactivate the dropdown list for col  if str2double(app.↙
YaxisUseDataDropDown.Value) == 1or line depending on the number of files uploaded↙
and the selected plotting style.
2995            value = str2double(app.YaxisUseDataDropDown.Value);
2996            isSingleStation = length(app.GaugeDirectorylist) == 1;
2997            isMultipleStations = length(app.GaugeDirectorylist) > 1;
2998            plotAllData = app.PlotalldatainonegraphButton.Value;
2999            plotSeparately = app.PlotseparatelyButton_4.Value;
3000
3001            if ~ismember(value, [3, 4, 5]) % Will only plot data from one↙
column
3002                if isSingleStation
3003                    setGaugeLineColorVisibility(true, false, false);
3004                elseif isMultipleStations
3005                    if plotAllData
3006                        setGaugeLineColorVisibility(false, true, true);
3007                    elseif plotSeparately
3008                        setGaugeLineColorVisibility(true, false, false,↙
false);
3009                    end
3010                end
3011            else % Use columns 3, 4, 5 to plot in Y-axis
3012                if isSingleStation
3013                    setGaugeLineColorVisibility(true, false, false);
3014                elseif isMultipleStations
3015                    if plotAllData
3016                        setGaugeLineColorVisibility(false, true, true);
3017                    elseif plotSeparately
3018                        setGaugeLineColorVisibility(true, false, false,↙
false);
3019                    end
3020                end
3021            end
3022
3023            function setGaugeLineColorVisibility(lineColorVisible,↙
lineColorMultipleVisible, flipCheckBoxVisible, flipCheckBoxEnable)
3024                if nargin < 4
3025                    flipCheckBoxEnable = true; % Default to true if not↙
specified
3026                end
3027                app.GaugeLineColor.Visible = lineColorVisible;
3028                app.GaugeLineColor_multiple.Visible =↙
lineColorMultipleVisible;
3029                app.FlipCheckBox_3.Visible = flipCheckBoxVisible;
3030                app.FlipCheckBox_3.Enable = flipCheckBoxEnable;
3031            end
3032 %             if ~ismember(str2double(app.YaxisUseDataDropDown.Value), [3, 4,↙
5])%Will only plot data from one column
3033 %                 if length(app.GaugeDirectorylist) == 1 %one station, use↙
dropdown for selecting one color
3034 %                     app.GaugeLineColor.Visible = "on";
3035 %                     app.GaugeLineColor_multiple.Visible = "off";
3036 %                     app.FlipCheckBox_3.Visible = "off";
3037 %                 elseif length(app.GaugeDirectorylist) > 1  && app.↙
PlotalldatainonegraphButton.Value %multiple stations, use dropdown for selecting a↙
set of color
3038 %                     app.GaugeLineColor.Visible = "off";
3039 %                     app.GaugeLineColor_multiple.Visible = "on";
```

```
3040 %                     app.FlipCheckBox_3.Visible = "on";
3041 %                 elseif length(app.GaugeDirectorylist) > 1  && app.↙
PlotseparatelyButton_4.Value
3042 %                     app.FlipCheckBox_3.Enable = "off";
3043 %                     app.GaugeLineColor.Visible = "on";
3044 %                     app.GaugeLineColor_multiple.Visible = "off";
3045 %                 end
3046 %             else %Use columns 3,4,5 to plot in Y-axis
3047 %                 if length(app.GaugeDirectorylist) == 1 %one station, use↙
dropdown for selecting one color
3048 %                     app.GaugeLineColor.Visible = "on";
3049 %                     app.GaugeLineColor_multiple.Visible = "off";
3050 %                     app.FlipCheckBox_3.Visible = "off";
3051 %                 elseif length(app.GaugeDirectorylist) > 1  && app.↙
PlotalldatainonegraphButton.Value %multiple stations, use dropdown for selecting a↙
set of color
3052 %                     app.GaugeLineColor.Visible = "off";
3053 %                     app.GaugeLineColor_multiple.Visible = "on";
3054 %                     app.FlipCheckBox_3.Visible = "on";
3055 %                 elseif length(app.GaugeDirectorylist) > 1  && app.↙
PlotseparatelyButton_4.Value
3056 %                     app.FlipCheckBox_3.Enable = "off";
3057 %                     app.GaugeLineColor.Visible = "on";
3058 %                     app.GaugeLineColor_multiple.Visible = "off";
3059 %                 end
3060 %             end
3061
3062             % Activate/deactivate the options for the plot legend and the↙
option to save the sta_ file as a .txt file.
3063             if app.PlotseparatelyButton_4.Value && length(app.↙
GaugeDirectorylist) == 1
3064                 app.legendfirsttext.Enable = "off";
3065                 app.LegendSize.Enable = "off";
3066                 app.LocationDropDown.Enable = "off";
3067                 app.txtCheckBox.Enable = "on";
3068             elseif app.PlotseparatelyButton_4.Value && length(app.↙
GaugeDirectorylist) > 1
3069                 app.legendfirsttext.Enable = "on";
3070                 app.LegendSize.Enable = "on";
3071                 app.LocationDropDown.Enable = "on";
3072                 app.txtCheckBox.Enable = "on";
3073             elseif app.PlotalldatainonegraphButton.Value
3074                 app.legendfirsttext.Enable = "on";
3075                 app.LegendSize.Enable = "on";
3076                 app.LocationDropDown.Enable = "on";
3077                 app.txtCheckBox.Enable = "off";
3078             end
3079
3080         end
3081
3082         % Value changed function: GaugeLineColor_multiple
3083         function GaugeLineColor_multipleValueChanged(app, event)
3084
3085         end
3086
3087         % Value changed function: AutoSetCheckBox
3088         function AutoSetCheckBoxValueChanged(app, event)
3089             if app.AutoSetCheckBox.Value
```

```
3090                 app.Width.Enable = "off";
3091                 app.Height.Enable = "off";
3092             else
3093                 app.Width.Enable = "on";
3094                 app.Height.Enable = "on";
3095             end
3096         end
3097 
3098         % Callback function
3099         function CloseFiguresButtonPushed(app, event)
3100             close all %figures
3101         end
3102 
3103         % Value changed function: mp4CheckBox
3104         function mp4CheckBoxValueChanged(app, event)
3105             if app.mp4CheckBox.Value
3106                 app.FramerateEditField_2.Visible = "on";
3107                 app.FramerateEditField_2Label.Visible="on";
3108             else
3109                 app.FramerateEditField_2.Visible = "off";
3110                 app.FramerateEditField_2Label.Visible="off";
3111             end
3112         end
3113 
3114         % Callback function
3115         function CloseFiguresButton_2Pushed(app, event)
3116             close all
3117         end
3118 
3119         % Selection changed function: ButtonGroup_11
3120         function ButtonGroup_11SelectionChanged(app, event)
3121             % Enable the tif and animation options in the 'Save Map' section↙
when 'Plot all data in one figure' is selected
3122             if app.PlotseparatelyButton_2.Value
3123                 app.mp4CheckBox.Enable = "on";
3124                 app.tifCheckBox_2.Enable = "on";
3125                 app.FramerateEditField_2.Enable = "on";
3126                 app.FramerateEditField_2Label.Enable = "on";
3127             elseif app.PlotalldatainonefigureButton.Value
3128                 app.mp4CheckBox.Enable = "off";
3129                 app.tifCheckBox_2.Enable = "off";
3130                 app.FramerateEditField_2.Enable = "off";
3131                 app.FramerateEditField_2Label.Enable = "off";
3132             end
3133         end
3134 
3135         % Value changed function: AutoSetCheckBox_2
3136         function AutoSetCheckBox_2ValueChanged(app, event)
3137             %Disable the Width and Height textboxes
3138             if app.AutoSetCheckBox_2.Value
3139                 app.Width_2.Enable = "off";
3140                 app.Height_2.Enable = "off";
3141             else
3142                 app.Width_2.Enable = "on";
3143                 app.Height_2.Enable = "on";
3144             end
3145 
3146         end
```

```
3147
3148        % Button pushed function: CloseFiguresButton_4
3149        function CloseFiguresButton_4Pushed2(app, event)
3150            close all
3151        end
3152
3153        % Button pushed function: Button_19
3154        function Button_19Pushed(app, event)
3155
3156            % Find U-vector files to import from a directory
3157            [files, path] = uigetfile('*.*', 'Select the files',↙
'MultiSelect', 'on');
3158
3159            % Convert path and files to strings once
3160            pathStr = string(path);
3161            filesStr = string(files);
3162
3163            if exist(pathStr, 'dir')
3164                cd(fullfile(pathStr));
3165
3166                % Collect all the files
3167                allfiles = fullfile(pathStr + filesStr); % fullfile is used to↙
handle folder names with spaces
3168                if ~(ismcc || isdeployed)
3169                    addpath(genpath(fullfile(pathStr)));
3170                end
3171
3172                % Sort the files by snapshot timing, and display the list of↙
filenames in the app's textbox
3173                sortedFiles = sort(filesStr);
3174                app.Uvectorfullfile = sort(allfiles);
3175                app.vectorpath = pathStr;
3176                app.Ufiles = sortedFiles;
3177                app.FileNAME5 = sortedFiles;
3178                app.FileTextArea_3.Value = sortedFiles;
3179                app.FileTextArea_3.FontColor = 'k';
3180                app.FileTextArea_3.BackgroundColor = 'w';
3181            end
3182
3183            app.MinBarValue.Enable = "on";
3184            app.MaxBarValue.Enable = "on";
3185
3186            ArrowSection = app.Panel_14.Children; % Get all the items in the↙
Arrow section
3187            if contains(app.FileNAME5, "umax") % if the file is for plotting↙
the maximum velocity, do not find the corresponding V-vector file
3188                % Disable the Arrow Section
3189                set(ArrowSection, 'Enable', 'off');
3190                app.PlotvectorsCheckBox.Value = false; % Uncheck the box
3191
3192                % Change default value of the colormap and the bar limits
3193                app.BackgroundMapColorDropDown.Value = 'blue - purple';
3194                app.MaxBarValue.Value = 6;
3195                app.InterpolationDivisionEditField_2.Value = 4;
3196                app.InterpolationDivisionEditField_2.Enable = "on";
3197                app.InterpolationDivisionEditFieldLabel_2.Enable = "on";
3198                app.BackgroundMapColorDropDown.Enable = "on";
3199                app.FlipCheckBox_4.Enable = "on";
```

```
3200            else
3201                % Enable the Arrow Section
3202                set(ArrowSection, 'Enable', 'on');
3203
3204                % Find the equivalent V Vector files, display the list of filenames in the app's textbox
3205                vfilenames = regexprep(sortedFiles, "u", "v", 'once'); % replaces only the first letter "u" in each of the filenames
3206                app.Vvectorfullfile = fullfile(pathStr + vfilenames);
3207                app.Vvectorfilenames = vfilenames;
3208                app.FileTextArea_4.Value = vfilenames;
3209                app.FileTextArea_4.FontColor = 'k';
3210                app.FileTextArea_4.BackgroundColor = 'w';
3211                app.BackgroundMapColorDropDown.Enable = "on";
3212                app.InterpolationDivisionEditField_2.Enable = "on";
3213                app.InterpolationDivisionEditFieldLabel_2.Enable = "on";
3214            end
3215
3216            % Find corresponding 'eta' files when 'eta' is selected as the basemap option
3217            ETAfiles = regexprep(sortedFiles, "u", "eta", 'once'); % change 'u' to 'eta' in the filename
3218            app.ETAfullfile = fullfile(pathStr + ETAfiles);
3219            app.ETAfilenames = ETAfiles;
3220
3221            % Find corresponding 'hmax' files when 'hmax' is selected as the basemap option
3222            HMAXfiles = regexprep(sortedFiles, "u", "hmax", 'once'); % change 'u' to 'hmax' in the filename
3223            app.HMAXfullfile = fullfile(pathStr + HMAXfiles);
3224            app.HMAXfilenames = HMAXfiles;
3225
3226            % Read the first file to extract its matrix size, which will be used to set up the map boundary limit
3227            if exist(app.Uvectorfullfile{1}, 'file') == 2
3228                % Read the matrix from the file
3229                f = readmatrix(app.Uvectorfullfile{1});
3230                % Setup of the coordinates
3231                [row, col] = size(f);
3232                app.col1vel = col - 1;
3233                app.row1vel = row - 1;
3234            end
3235
3236        end
3237
3238        % Callback function
3239        function Button_20Pushed(app, event)
3240
3241        end
3242
3243        % Button pushed function: GENERATEButton_2
3244        function GENERATEButton_2Pushed(app, event)
3245            %Defining Parameters
3246            close all
3247            clear depthVM Uvector Vvector VMdata
3248
3249            %-----------------------
3250            %------ARROW COLOR------
```

```
3251            %-----------------------
3252            quivercolorOptions = containers.Map({'Black', 'Dark gray', 'Medium gray', 'Light gray', 'Red', 'Green', 'Blue', 'Yellow', 'Cyan', 'Magenta', 'White'}, {'k', '[0.4 0.4 0.4]', '[0.7 0.7 0.7]', '[0.8 0.8 0.8]', 'r', 'g', 'b', 'y', 'c', 'm', 'w'});
3253            quivercolor = quivercolorOptions(app.QuiverColorDropDown.Value);
3254
3255
3256            %---------------------------
3257            %------LAND AREA COLOR------
3258            %---------------------------
3259            LandAreaColorOptions = containers.Map({'Black', 'Dark gray', 'Medium gray', 'Light gray', 'Red', 'Green', 'Blue', 'Yellow', 'Cyan', 'Magenta', 'White'}, {'k', '[0.4 0.4 0.4]', '[0.7 0.7 0.7]', '[0.8 0.8 0.8]', 'r', 'g', 'b', 'y', 'c', 'm', 'w'});
3260            LandAreaColor2 = LandAreaColorOptions(app.LandColor.Value);
3261
3262
3263            %---------------------------
3264            %------BASEMAP COLORMAP------
3265            %---------------------------
3266            %List of the  colormap items in the dropdown list
3267            keys = {'autumn', 'bone', 'colorcube', 'cool', 'copper', 'flag', 'gray', 'hot', 'hsv', 'jet', ...
3268                'parula', 'pink', 'prism', 'spring', 'summer', 'turbo', 'winter', 'blue', 'blue - green', ...
3269                'blue - purple', 'green - blue', 'greens', 'gray', 'oranges', 'orange - red', 'purple - blue', ...
3270                'purple - blue - green', 'purple - red', 'purples', 'red - purple', 'reds', 'yellow - green', ...
3271                'yellow - green - blue', 'yellow - orange - brown', 'yellow - orange - red', 'brown - teal', ...
3272                'pink - light green', 'purple - green', 'purple - orange', 'red - blue', 'red - gray', ...
3273                'red - yellow - blue', 'red - yellow - green', 'spectral', 'accent', 'dark 2', 'paired', ...
3274                'pastel 1', 'pastel 2', 'set 1', 'set 2', 'set 3', '--- MATLAB default ----', '---  CBREWER 2 ---', ...
3275                '< sequential >', '< divergent >', '< qualitative >'};
3276
3277            %The equivalent colormap code
3278            keysColor = {'autumn', 'bone', 'colorcube', 'cool', 'copper', 'flag', 'gray', 'hot', 'hsv', 'jet', ...
3279                'parula', 'pink', 'prism', 'spring', 'summer', 'turbo', 'winter','Blues','BuGn','BuPu','GnBu','Greens', ...
3280                'Greys','Oranges','OrRd','PuBu','PuBuGn', ...
3281 'PuRd','Purples','RdPu','Reds','YlGn','YlGnBu','YlOrBr','YlOrRd','BrBG','PiYG','PRGn','PuOr','RdBu','RdGy', ...
3282                'RdYlBu','RdYlGn','Spectral','Accent', ...
3283 'Dark2','Paired','Pastel1','Pastel2','Set1','Set2','Set3','parula','parula','parula','parula','parula'};
3284
3285            %The colorbrewer
3286            keysBrewer = {'none', 'none', 'none', 'none', 'none', 'none', 'none', 'none', 'none', 'none', ...
3287                'none', 'none', 'none', 'none', 'none', 'none', 'none', 'seq',
```

```
'seq', ...
3288                'seq', 'seq', 'seq', 'seq', 'seq', 'seq', ...
3289                'seq', 'seq', 'seq', 'seq', 'seq', ...
3290                'seq', 'seq', 'seq', 'seq', 'seq', 'div', 'div', 'div', ...
3291                'div', 'div', 'div', 'div', 'div', ...
3292                'div', 'qual', 'qual', 'qual', ...
3293                'qual', 'qual', 'qual', 'qual', 'qual', ...
3294                'none', 'none', 'none','none', 'none'};
3295
3296            % Match dropdown list items to their corresponding colors and
ColorBrewer
3297            colorMap = containers.Map(keys, keysColor);
3298            colorBrewer = containers.Map(keys, keysBrewer);
3299
3300            %Set up the colormap
3301            datacolor2  = colorMap(app.BackgroundMapColorDropDown.Value);
3302            datacolorbrewer2 = colorBrewer(app.BackgroundMapColorDropDown.
Value);
3303
3304
3305            %-------------------------------
3306            %------BATHYMETRY CONTOURS------
3307            %-------------------------------
3308            %Line Colours
3309            ContourColor2Options = containers.Map({'Black', 'Dark gray',
'Medium gray', 'Light gray', 'Red', 'Green', 'Blue', 'Yellow', 'Cyan', 'Magenta',
'White'}, {'k', '[0.4 0.4 0.4]', '[0.7 0.7 0.7]', '[0.8 0.8 0.8]', 'r', 'g', 'b',
'y', 'c', 'm', 'w'});
3310            ContourColor2 = ContourColor2Options(app.ColorDropDown_4.Value);
3311
3312            %Line style
3313            ContourLineStyle2Options = containers.Map({'Solid', 'Dashed',
'Dotted', 'Dash-dotted'}, {'-', '--', ':', '-.'});
3314            ContourLineStyle2 = ContourLineStyle2Options(app.StyleDropDown_3.
Value);
3315
3316
3317            %------------------
3318            %------GAUGES------
3319            %------------------
3320            %Marker Style
3321            GaugeMarkerOptions = containers.Map({'none', 'o', '+', '*', '.',
'x', '-', '|', '^', 'v', '>', '<', 'diamond', 'hexagram', 'pentagram', 'square'},
...
3322                {'none', 'o', '+', '*', '.', 'x', '-', '|', '^', 'v', '>',
'<', 'diamond', 'hexagram', 'pentagram', 'square'});
3323            GaugeMarker2 = GaugeMarkerOptions(app.gaugemarkerVelocityTab.
Value);
3324
3325            %Marker Color
3326            GaugeColor2Options = containers.Map({'Black', 'Dark gray', 'Medium
gray', 'Light gray', 'Red', 'Green', 'Blue', 'Yellow', 'Cyan', 'Magenta', 'White'},
{'k', '[0.4 0.4 0.4]', '[0.7 0.7 0.7]', '[0.8 0.8 0.8]', 'r', 'g', 'b', 'y', 'c',
'm', 'w'});
3327            GaugeColor2 = GaugeColor2Options(app.ColorDropDown_7.Value);
3328
3329            %Label - horizontal alignment
3330            HorzLabel2Options = containers.Map({'Centre', 'Right', 'Left'},
```

```
{'center', 'left', 'right'});
3331            HorzLabel2 = HorzLabel2Options(app.CoastlinecolourDropDown_10.↙
Value);
3332
3333            %Label - vertical alignment
3334            VertLabel2Options = containers.Map({'Centre', 'Top', 'Bottom'},↙
{'middle', 'bottom', 'top'});
3335            VertLabel2 = VertLabel2Options(app.CoastlinecolourDropDown_11.↙
Value);
3336
3337
3338            %--------------------------------------
3339            %------SAVE MAP: Output Directory------
3340            %--------------------------------------
3341            %Check if the output directory has been manually set by the user
3342            if  app.OutputDirectoryEditField_3.Value == string(app.↙
WORKFOLDER4)
3343                app.MainDirectoryVector= fullfile(app.WORKFOLDER3);
3344                app.FigureDirectoryVector = fullfile(app.WORKFOLDER4);
3345            else % Set the default directory to the Desktop
3346                if ismac %macOS
3347                    defaultDir = fullfile(getenv('HOME'), 'Desktop');
3348                elseif ispc %Windows
3349                    defaultDir = fullfile(getenv('USERPROFILE'), 'Desktop');
3350                else %Others
3351                    defaultDir = pwd;
3352                end
3353
3354                %Create 'OUTPUT_FILES' folder and 'Figures' subfolder
3355                Dir0 = fullfile(defaultDir, 'OUTPUT_FILES');
3356                Dir1 = fullfile(defaultDir, 'OUTPUT_FILES/Figures');
3357                app.MainDirectoryVector= Dir0;
3358                app.FigureDirectoryVector = Dir1;
3359                if ~exist(Dir0, 'dir')
3360                    mkdir(Dir0);
3361                end
3362                if ~exist(Dir1, 'dir')
3363                    mkdir(Dir1);
3364                end
3365
3366                %Display the path directory of the 'Figures' subfolder in the↙
textbox
3367                app.OutputDirectoryEditField_3.Value = deblank(string(Dir1));
3368                app.OutputDirectoryEditField_3.FontColor = 'k';
3369            end
3370
3371
3372
3373            %Create animation/ creating and opening the file
3374            if app.mp4CheckBox_2.Value
3375                vidObj = VideoWriter(fullfile(char(app.FigureDirectoryVector),↙
'animation_vector.mp4'), 'MPEG-4');
3376                vidObj.FrameRate = app.FramerateEditField_2.Value;
3377                open(vidObj);
3378            end
3379
3380            %----------------------
3381            %------LOG REPORT------
```

```
3382            %----------------------
3383            %Create 'Log_Files' subfolder in 'OUTPUT_FILES'
3384            LogFolder = fullfile(app.MainDirectoryVector, 'Log_Files');
3385            if ~exist(LogFolder, 'dir')
3386                mkdir(LogFolder);
3387            end
3388            %Create the text file
3389            filePath = fullfile(LogFolder, 'Log_Report_VelocityMap.txt');
3390            fileId = fopen(filePath ,'w');
3391
3392            % Template for the default section of the log report (includes
information from the 'Input Data' section)
3393            headerTitle = '******************** LOG REPORT
******************** ';
3394            timestamp = datestr(now, 'yyyy-mm-dd HH:MM:SS');
3395            endHeaderLine =
'************************************************';
3396            footerSeparator =
'________________________________________________';
3397            logContent = [
3398                sprintf('%s\n', headerTitle), ...
3399                sprintf('%-30s%-s\n', 'Map Type:', 'Velocity Map'), ...
3400                sprintf('%-30s%-s\n', 'Timestamp:', timestamp), ...
3401                sprintf('%s\n\n', endHeaderLine), ...
3402                sprintf('%-30s%-s\n', 'Southwest Corner:', sprintf('Long: %f
Lat: %f', app.LongitudeEditField_2.Value, app.LatitudeEditField_2.Value)), ...
3403                sprintf('%-30s%-s\n', 'Grid Size:', sprintf('x: %d y: %d',
app.gridX_2.Value, app.gridY_2.Value)), ...
3404                sprintf('%-30s%-s\n', 'Simulation Time Start:', sprintf('%d
sec', app.StartTime2.Value)), ...
3405                sprintf('%-30s%-s\n', 'Simulation Time Interval:', sprintf('%d
sec', app.TotalSimuilationTimesecEditField_4.Value)), ...
3406                sprintf('%s\n\n', footerSeparator)
3407                ];
3408            fprintf(fileId, '%s', logContent);
3409
3410
3411            %------LOG REPORT--------------------
3412            %------Input Data: Import Files------
3413            %------------------------------------
3414            %Check loaded files in the U vector input form
3415            patterns = {'u_', 'umax_', 'umean_'};  %Check each loaded files if
they contain "u"
3416            matches1 = false(size(app.FileNAME5));
3417            nonMatchingFiles = ~contains(string(app.FileNAME5), patterns);
3418            for i = 1:numel(patterns)
3419                pattern = patterns{i};
3420                matches1 = matches1 | contains(string(app.FileNAME5),
pattern);
3421            end
3422
3423            if any(nonMatchingFiles)
3424                fprintf(fileId, '%-30s%-s\n', 'U VECTOR:', 'Use filenames with
''u_'', ''umax_'', or ''umean_'' initials');
3425                fprintf(fileId, '%-30s%-s\n', ' ' , 'Incorrect files are
loaded:');
3426                fprintf(fileId, '%-30s%-s\n', ' ', join(app.FileNAME5
(nonMatchingFiles) , ', ')); % List all files that do not start with 'u' in the
```

```
filename
3427                 fprintf(fileId, '%-30s%-s\n\n', ' ', 'No vectors are↙
plotted');
3428             elseif all(contains(string(app.FileNAME5),  patterns)) %If all↙
files are in U vector format
3429                 %Check if V vector files exists in the same directory as the U↙
vectors
3430                 missingFiles = {};
3431                 for i = 1:length(app.Vvectorfullfile) % Check each filename in↙
the list
3432                     filePath = fullfile(app.Vvectorfullfile{i});
3433                     if exist(filePath, 'file') ~= 2
3434                         missingFiles = [missingFiles, app.Vvectorfilenames↙
{i}]; %Make a list all corresponding V vector files that are not found
3435                     end
3436                 end
3437                 if ~isempty(missingFiles)
3438                     fprintf(fileId,'%-30s%-s\n', 'V VECTOR :','Missing Files.↙
Ensure they are in the same folder as the U vector files:');
3439                     fprintf(fileId, '%-30s%-s\n\n', ' ', strjoin(missingFiles,↙
', '));
3440                 end
3441             end
3442
3443                 %------LOG REPORT-----------------------
3444                 %------Input Data: Southwest Corner------
3445                 %---------------------------------------
3446                 %Check if the values have not changed and add an error note to↙
the log report
3447                 if app.LongitudeEditField_2.Value == 0 && app.↙
LatitudeEditField_2.Value == 0
3448                     fprintf(fileId, '%-30s%-s\n\n', 'SOUTHWEST CORNER:','Both↙
latitude and longitude values are zeroes');
3449                 elseif app.LongitudeEditField_2.Value == 0 && app.↙
LatitudeEditField_2.Value ~= 0
3450                     fprintf(fileId, '%-30s%-s\n\n', 'SOUTHWEST↙
CORNER:','Longitude value is zero');
3451                 elseif app.LatitudeEditField_2.Value ~= 0 && app.↙
LatitudeEditField_2.Value == 0
3452                     fprintf(fileId, '%-30s%-s\n\n', 'SOUTHWEST↙
CORNER:','Latitude value is zero');
3453                 end
3454
3455                 %------LOG REPORT-----------------
3456                 %------Input Data: Grid Size------
3457                 %---------------------------------
3458                 %Check if the values have not changed and add an error note to↙
the log report
3459                 if app.gridX_2.Value == 0 && app.gridY_2.Value == 0
3460                     fprintf(fileId, '%-30s%-s\n\n', 'GRID SIZE:','Both x and y↙
values are zeroes');
3461                 elseif app.gridX.Value == 0 && app.gridY.Value ~= 0
3462                     fprintf(fileId, '%-30s%-s\n\n', 'GRID SIZE:','X value is↙
zero');
3463                 elseif app.gridX.Value ~= 0 && app.gridY.Value == 0
3464                     fprintf(fileId, '%-30s%-s\n\n', 'GRID SIZE:','Y value is↙
zero');
3465                 end
```

```
3466
3467                 %------LOG REPORT-----------------------
3468                 %------Input Data: Load Bathymetry------
3469                 %---------------------------------------
3470                 %Check if the matrix size of the loaded bathymetry file↙
matches the imported eta/hmax file
3471                 if matches(app.FileTextArea_3.Value,string(app.FileNAME5)) %↙
Check if the imported V files match the names in the textbox
3472                     if matches(app.DepthFileEditField_2.Value,string(app.↙
bathymetryname2)) %Check if the imported bathymetry matches the name in the textbox
3473                         [rowbath,colbath] = size(app.bathymetryname2);
3474                         if   app.row1vel ~= rowbath
3475                             fprintf(fileId, '%-30s%-s\n\n', 'MATRIX↙
ROWS:','Rows in U-vector file and bathymetry do not match');
3476                         end
3477                         if app.col1vel ~= colbath
3478                             fprintf(fileId, '%-30s%-s\n\n', 'MATRIX↙
COLUMNS:','Columns in U-vector file and bathymetry do not match');
3479                         end
3480                     end
3481                 end
3482
3483                 if app.DepthFileEditField_2.Value == "0" || ~matches(app.↙
DepthFileEditField_2.Value,string(app.bathymetryname2)) % Add a log note that no↙
bathymetry is loaded, so the features to edit land area and bathymetry contours are↙
unavailable.
3484                     fprintf(fileId, '%-30s%-s\n\n', 'BATHYMETRY:', 'Empty.↙
Land color and bathymetry contours are not applicable.');
3485                     app.DepthFileEditField_2.Value = "NO FILE";
3486                     app.DepthFileEditField_2.FontColor = 'w';
3487                     app.DepthFileEditField_2.BackgroundColor = 'r';
3488                 end
3489
3490                 %Check if the bathymetry is has the correct file format
3491                 if app.bathymetrycheck == 1
3492                     fprintf(fileId, '%-30s%-s\n\n', 'BATHYMETRY:', 'File must↙
be .txt, .tif or mask_');
3493                 end
3494
3495                 %------LOG REPORT-------------------------
3496                 %------Basemap and Overlays: Basemap------
3497                 %-----------------------------------------
3498                 if app.MinBarValue.Value > app.MaxBarValue.Value
3499                     fprintf(fileId, '%-30s%-s\n', 'BASEMAP COLORBAR:','Max↙
value must exceed min value');
3500                     fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved: Max and min↙
values are swapped');
3501                 end
3502
3503                 %------LOG REPORT---------------------------------------
3504                 %------Basemap and Overlays: Bathymetry Contours------
3505                 %-------------------------------------------------------
3506                 if app.PlotBathymetryContoursCheckBox.Value
3507                     if matches(app.DepthFileEditField_2.Value ,string(app.↙
bathymetryname2))
3508                         % Check if the minimum and maximum values are in the↙
correct order
3509                         if app.MinimumEditField_2.Value > app.↙
```

```
MaximumEditField_2.Value
3510                            temp = app.MaximumEditField_2.Value;
3511                            app.MaximumEditField_2.Value = app.↙
MinimumEditField_2.Value;
3512                            app.MinimumEditField_2.Value = temp;
3513                            fprintf(fileId, '%-30s%-s\n', 'BATHYMETRY↙
CONTOURS:','Max value must exceed min value');
3514                            fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved:↙
Values are swapped');
3515                        end
3516
3517                        %Check the interval value is within the limit
3518                        maxMinDifference = app.MaximumEditField_2.Value - app.↙
MinimumEditField_2.Value;
3519                        if app.IntervalEditField_6.Value > maxMinDifference
3520                            app.IntervalEditField_6.Value = maxMinDifference /↙
2;
3521                            fprintf(fileId, '%-30s%-s\n', 'BATHYMETRY CONTOUR↙
INTERVAL:','Input must be less than max-min difference');
3522                            fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved:↙
Value adjusted to half the max-min difference');
3523                        end
3524
3525                        if app.AddLabelCheckBox_2.Value %Add labels
3526                            %Check if the label interval is divisible by the↙
value provided in the 'Range' section
3527                            if mod(app.IntervalEditField_8.Value, app.↙
IntervalEditField_6.Value) ~= 0
3528                                app.IntervalEditField_8.Value = app.↙
IntervalEditField_6.Value;
3529                                fprintf(fileId, '%-30s%-s\n', 'BATHYMETRY↙
CONTOUR LABEL:','Value must be divisible by the Interval value in the Range↙
section');
3530                                fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved:↙
Value adjusted to be the same as the Interval Value in the Range section');
3531                            end
3532                        end
3533                    end
3534                end
3535
3536                %------LOG REPORT-----------------------
3537                %------Basemap and Overlays: Gauges------
3538                %---------------------------------------
3539                if app.PlotGaugesCheckBox.Value && ~matches(app.↙
FileEditField_2.Value ,string(app.GAUGEFILE3))
3540                    fprintf(fileId, '%-30s%-s\n\n', 'GAUGES:', 'Empty. No↙
virtual gauges plotted');
3541                    app.FileEditField_2.Value = "NO FILE";
3542                    app.FileEditField_2.FontColor = 'w';
3543                    app.FileEditField_2.BackgroundColor = 'r';
3544                end
3545
3546                %------LOG REPORT---------------------------
3547                %------General Layout: Boundary Limits------
3548                %-------------------------------------------
3549                if app.WestEditField_2.Value > app.EastEditField_2.Value
3550                    fprintf(fileId, '%-30s%-s\n', 'BOUNDARY LIMIT:','East↙
boundary must be greater than West boundary');
```

```
3551                     fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved: Values are↙
swapped');
3552                 elseif app.EastEditField_2.Value == app.WestEditField_2.Value
3553                     fprintf(fileId, '%-30s%-s\n', 'BOUNDARY LIMIT:', 'East and↙
West values are the same');
3554                     fprintf(fileId, '%-30s%-s\n\n', 'Resolved:', 'The default↙
values are used');
3555                 end
3556
3557                 if app.SouthEditField_2.Value > app.NorthEditField_2.Value
3558                     fprintf(fileId, '%-30s%-s\n', 'BOUNDARY LIMIT:','North↙
boundary must be greater than South boundary');
3559                     fprintf(fileId, '%-30s%-s\n\n', ' ','Resolved: Values are↙
swapped');
3560                 elseif  app.NorthEditField_2.Value == app.SouthEditField_2.↙
Value
3561                     fprintf(fileId, '%-30s%-s\n', 'BOUNDARY LIMIT:', 'North↙
and South values are the same');
3562                     fprintf(fileId, '%-30s%-s\n\n', 'Resolved:', 'The default↙
values are used');
3563                 end
3564
3565                 fclose(fileId); %Close the log report file
3566
3567                 %% PROCESS THE INPUT FILES AND VALUES, THEN GENERATE THE↙
FIGURES
3568                 %---------------------------------
3569                 %------READ THE INPUT FILES------
3570                 %---------------------------------
3571                 % Set up the bathymetry data
3572                 bathymetryFileExists = exist(app.bathymetryinputdata2, 'file')↙
== 2;
3573                 if bathymetryFileExists
3574                     depthVM = app.bathymetrydata2;
3575                     if app.HorizontalCheckBox_2.Value
3576                         depthVM = fliplr(depthVM);
3577                     end
3578                     if app.VerticalCheckBox_2.Value
3579                         depthVM = flipud(depthVM);
3580                     end
3581                 end
3582
3583                 FileLength1 = length(app.Uvectorfullfile);
3584                 patterns = {'u_', 'umax_', 'umean_'};
3585
3586                 %---------------------------------------
3587                 %------PLOT ALL DATA IN ONE FIGURE------
3588                 %---------------------------------------
3589                 if app.PlotalldatainonefigureButton_2.Value
3590
3591                     %Create the figure
3592                     f = figure(1);
3593                     f.Visible = 'off';
3594
3595                     %Formatting the subplots
3596                     if FileLength1 == 1
3597                         plotCol = 1;
3598                         plotRow = 1;
```

```
3599                    elseif FileLength1 > 1 && FileLength1 <= 10
3600                        plotCol = 2;
3601                        plotRow = ceil(FileLength1/2);
3602                    elseif FileLength1 > 10
3603                        plotCol = 3;
3604                        plotRow = ceil(FileLength1/3);
3605                    end
3606
3607                    %---------------------------
3608                    %------PLOT SEPARATELY------
3609                    %---------------------------
3610                elseif app.PlotseparatelyButton_3.Value
3611                    f = figure();
3612                    f.Visible = 'off';
3613                end
3614
3615                %--------------------------------
3616                %------PRE-SET MAP OPTIONS-------
3617                %--------------------------------
3618                %Allocate memory for the vectors
3619                Uvector = NaN(app.row1vel, app.col1vel);
3620                Vvector = NaN(app.row1vel, app.col1vel);
3621                colorbarHandles = [];
3622
3623                %------Basemap Colormap-------
3624                if strcmp(datacolorbrewer2,'none') %Using default matlab↙
colormap options
3625                    datacolor3 = evalin('base',datacolor2);
3626                else %Using cbrewer2
3627                    datacolor3 = cbrewer2(datacolorbrewer2,datacolor2,app.↙
InterpolationDivisionEditField_2.Value,'pchip');
3628                end
3629
3630                %------Basemap Colorbar-------
3631                colormapMin = app.MinBarValue.Value;
3632                colormapMax = app.MaxBarValue.Value;
3633                %Options for label
3634                if app.BathymetryButton.Value
3635                    colorbartitle = 'Bathymetry(m)';
3636                elseif app.etaButton.Value
3637                    colorbartitle = '\eta (m)';
3638                elseif app.hmaxButton.Value
3639                    colorbartitle = 'Maximum \eta (m)';
3640                elseif app.VelocityButton.Value
3641                    if contains(string(app.Ufiles(i)),'umax')
3642                        colorbartitle = 'Maximum Velocity (m/s)';
3643                    else
3644                        colorbartitle = 'Velocity (m/s)';
3645                    end
3646                elseif app.VorticityButton.Value
3647                    colorbartitle = 'Vorticity (s^{-1})';
3648                end
3649
3650                %Check if the minimum and maximum values are in the correct↙
order; if not, swap them
3651                if app.BathymetryButton.Value
3652                    if -app.MaxBarValue.Value > -app.MinBarValue.Value %Min↙
and Max values of the colorbar limit are swapped
```

```
3653                        %Swap the values
3654                        temp = app.MaxBarValue.Value;
3655                        app.MaxBarValue.Value = app.MinBarValue.Value;
3656                        app.MinBarValue.Value = temp;
3657                    end
3658                else
3659                    if app.MaxBarValue.Value < app.MinBarValue.Value %Min and↙
Max values of the colorbar limit are swapped
3660                        %Swap the values
3661                        temp = app.MaxBarValue.Value;
3662                        app.MaxBarValue.Value = app.MinBarValue.Value;
3663                        app.MinBarValue.Value = temp;
3664                    end
3665                end
3666
3667                %------Boundary limit-------
3668                eastValue = app.EastEditField_2.Value;
3669                westValue = app.WestEditField_2.Value;
3670
3671                if eastValue < westValue
3672                    % Swap West and East input values if the West value is↙
greater than the East
3673                    xMin_VM = eastValue;
3674                    xMax_VM = westValue;
3675                    app.EastEditField_2.Value = xMax_VM;
3676                    app.WestEditField_2.Value = xMin_VM;
3677                elseif eastValue == westValue
3678                    % Set the limits to the default values
3679                    app.EastEditField_2.Value = max(app.x);
3680                    app.WestEditField_2.Value = min(app.x);
3681                    xMin_VM = app.WestEditField_2.Value;
3682                    xMax_VM = app.EastEditField_2.Value;
3683                else
3684                    xMin_VM = westValue;
3685                    xMax_VM = eastValue;
3686                end
3687
3688                northValue = app.NorthEditField_2.Value;
3689                southValue = app.SouthEditField_2.Value;
3690
3691                if northValue < southValue
3692                    % Swap North and South input values if the South value is↙
greater than the North
3693                    yMin_VM = northValue;
3694                    yMax_VM = southValue;
3695                    app.NorthEditField_2.Value = yMax_VM;
3696                    app.SouthEditField_2.Value = yMin_VM;
3697                elseif northValue == southValue
3698                    % Set the limits to the default values
3699                    app.NorthEditField_2.Value = max(app.y);
3700                    app.SouthEditField_2.Value = min(app.y);
3701                else
3702                    yMin_VM = southValue;
3703                    yMax_VM = northValue;
3704                end
3705
3706            %------Map Title-------
3707            totalSimulationTime = app.TotalSimuilationTimesecEditField 4. ↙
```

```
Value;
3708             startTime = app.StartTime2.Value;
3709             colorbarTextSize = app.maplabelsize.Value + 1;
3710             pat = digitsPattern;
3711
3712
3713                 %-----------------------------------------
3714                 %------Start of the loop for plotting-------
3715                 %-----------------------------------------
3716                 for i = 1:FileLength1
3717                     if all(contains(string(app.FileTextArea_3.Value),↙
patterns)) % Check if each file is a U vector
3718                         % Load the U vector files
3719                         if exist(app.Uvectorfullfile{i}, 'file') == 2
3720                             Uvector = readmatrix(app.Uvectorfullfile{i});
3721
3722 %                             %Remove values on land (negative values)
3723 %                             if bathymetryFileExists
3724 %                                 Uvector(depthVM <= 0) = NaN; % Set land↙
values to NaN or another appropriate value
3725 %                             end
3726                         end
3727
3728
3729                         % Load the V vector files
3730                         if ~contains(char(app.Ufiles{i}), 'umax') && exist↙
(app.Vvectorfullfile{i}, 'file') == 2
3731                             Vvector = readmatrix(app.Vvectorfullfile{i});
3732
3733                             %------Flip the Matrix-------
3734 %                             if app.HorizontalCheckBox_2.Value
3735 %                                 % Flip horizontally
3736 %                                 Vvector = fliplr(Vvector);
3737 %                             end
3738 %                             if  app.VerticalCheckBox_2.Value
3739 %                                 Vvector = flipud(Vvector);
3740 %                             end
3741
3742 %                             % Remove values on land if the bathymetry file↙
exists
3743 %                             if bathymetryFileExists
3744 %                                 Vvector(depthVM <= 0) = NaN; % Set land↙
values to NaN or another appropriate value
3745 %                             end
3746
3747                         end
3748
3749                         %----------------------------------------
3750                         %------PLOT ALL DATA IN ONE FIGURE------
3751                         %----------------------------------------
3752                         if app.PlotalldatainonefigureButton_2.Value
3753                             % Create a subplot in the specified position
3754                             gca(i) = subplot(plotRow, plotCol, i);
3755                             f.Visible = 'on';
3756
3757                             %--------------------------
3758                             %------PLOT SEPARATELY------
3759                             %--------------------------
```

```
3760                            elseif app.PlotseparatelyButton_3.Value
3761                                f = figure();
3762                                f.Visible = 'off';
3763                                gca(i) = subplot(1,1,1);
3764                            end
3765
3766                            %----------------------------------------
3767                            %------Basemap and Overlays: Basemap------
3768                            %----------------------------------------
3769                            %Check which option is selected for the basemap
3770                            clear VMdata
3771                            %Option 1: W Velocity
3772                            if app.VelocityButton.Value
3773                                if ~contains(char(app.Ufiles(i)),'umax') && exist↙
(app.Vvectorfullfile(i),'file')
3774                                    VMdata = sqrt(Uvector.^2 + Vvector.^2); %↙
Calculate the Z magnitude based on the magnitudes of the U and V vectors
3775                                elseif contains(char(app.Ufiles(i)),'umax')
3776                                    VMdata = Uvector;
3777                                end
3778
3779                                %Option 2: eta / sea surface displacement at time↙
t
3780                            elseif  app.etaButton.Value
3781                                VMdata = readmatrix(app.ETAfullfile(i));
3782
3783                                %Option 3: hmax / maximum wave height
3784                            elseif app.hmaxButton.Value
3785                                if exist(app.HMAXfullfile(i),"file")
3786                                    VMdata = readmatrix(app.HMAXfullfile(i));
3787                                end
3788
3789                                %Option 4: Bathymetry
3790                            elseif app.BathymetryButton.Value
3791                                if matches(app.DepthFileEditField_2.Value,string↙
(app.bathymetryname2))
3792                                    VMdata = -depthVM;
3793                                end
3794
3795                                %Option 5: Vorticity
3796                            elseif app.VorticityButton.Value
3797                                [VMdata, ~]=curl(app.xcoord,app.ycoord,Uvector, ↙
Vvector); %from simple_cases/rip_2d
3798                            end
3799
3800
3801                            %------Flip the Matrix-------
3802                            % Flip horizontally
3803                            if app.HorizontalCheckBox_2.Value && ~app.↙
BathymetryButton.Value
3804                                VMdata = fliplr(VMdata);
3805                            end
3806
3807                            % Flip vertically
3808                            if app.VerticalCheckBox_2.Value && ~app.↙
BathymetryButton.Value
3809                                VMdata = flipud(VMdata);
3810                            end
```

```
3811
3812                        %Remove land values from the selected basemap
3813                        try
3814                            if isprop(app, 'bathymetryinputdata2')
3815                                VMdata(depthVM<=0) = NaN;
3816                            end
3817                        end
3818
3819                        %---------------------
3820                        %------MAIN PLOT------
3821                        %---------------------
3822                        p = pcolor(app.xcoord,app.ycoord,VMdata);
3823                        shading interp
3824                        set(p, 'EdgeColor', 'none','FaceColor', 'interp');
3825                        set(gca,'Color', LandAreaColor2) %Set the background color for the land, %Adjust tick size label
3826                        hold on
3827
3828
3829                        %-----------------------------------------------------
3830                        %------Basemap and Overlays: Basemap Colormap-------
3831                        %-----------------------------------------------------
3832                        %Flip the colorbar
3833                        if app.FlipCheckBox_4.Value
3834                            colormap(gca(i), flipud(datacolor3));
3835                        else
3836                            colormap(gca(i), datacolor3);
3837                        end
3838
3839                        %-----------------------------------------------------
3840                        %------Basemap and Overlays: Basemap Colorbar-------
3841                        %-----------------------------------------------------
3842                        hold on
3843                        %Check if the minimum and maximum values are in the correct order; if not, swap them
3844                        if app.BathymetryButton.Value
3845                            caxis(gca(i), [-colormapMax,-colormapMin]); % Colorbar min and max values
3846                        else
3847                            caxis(gca(i), [colormapMin,colormapMax]); % Colorbar min and max values
3848                        end
3849
3850                        cb = colorbar;
3851                        colorbarHandles = [colorbarHandles, cb];
3852
3853                        if cb.FontSize ~= colorbarTextSize
3854                            cb.FontSize = colorbarTextSize; % size of the tick numbers
3855                        end
3856                        cb.Label.String = colorbartitle; % Set the colorbar title
3857
3858                        %-----------------------------------------------------
3859                        %------Basemap and Overlays: Bathymetry Contours----
3860                        %-----------------------------------------------------
3861                        if app.PlotBathymetryContoursCheckBox.Value && matches(app.DepthFileEditField_2.Value,string(app.bathymetryname2))
```

```
3862                                hold on
3863                                %Plot the contours
3864                                depth0 = depthVM;
3865                                kernel = ones(6) / 36; % Use averaging kernel to↙
make the contour smoother
3866                                depthfiltered = filter2(kernel, depth0);
3867                                [C1,h1] = contour(app.xcoord,app.ycoord,↙
depthfiltered,app.MinimumEditField_2.Value:app.IntervalEditField_6.Value:app.↙
MaximumEditField_2.Value,'EdgeColor',ContourColor2,'LineStyle',↙
ContourLineStyle2,'LineWidth',app.WidthEditField_2.Value);
3868                                %Add contour labels
3869                                if app.AddLabelCheckBox_2.Value
3870                                    hold on
3871                                    manualLabels3 = app.MinimumEditField_2.Value:↙
app.IntervalEditField_6.Value:app.MaximumEditField_2.Value;
3872                                    clabel(C1, h1, manualLabels3, 'FontSize', app.↙
LabelSizeCont.Value, 'Color', 'k');
3873                                    set(handle(h1), 'LabelSpacing', app.↙
SpacingEditField.Value);% Adjust the label spacing
3874                                end
3875                            end
3876
3877
3878                            %---------------------------------------
3879                            %------Basemap and Overlays: Gauges----
3880                            %---------------------------------------
3881                            hold on
3882                            if app.PlotGaugesCheckBox.Value && matches(app.↙
FileEditField_2.Value,string(app.GAUGEFILE3))
3883                                %Plot the points
3884                                gaugepoint = plot(app.longGauge2,app.latGauge2,↙
GaugeMarker2);
3885                                set(gaugepoint, 'MarkerSize', app.SizeEditField_2.↙
Value,'MarkerFaceColor', GaugeColor2,'MarkerEdgeColor', GaugeColor2); %Size and↙
color of the points
3886                                %Add labels
3887                                if app.AddLabelCheckBox.Value
3888                                    hold on
3889                                    for k = 1:length(app.latGauge2)
3890                                        gtxt = text(app.longGauge2(k)+ app.↙
GLspacing_2.Value,app.latGauge2(k,1),num2str(k),'FontSize',app.FontSizeEditField_2.↙
Value, 'HorizontalAlignment',HorzLabel2,...
3891                                            'VerticalAlignment',VertLabel2); %↙
GLspacing_2.Value is added to longitude so there's space between the point and the↙
label
3892                                        set(gtxt, 'Clipping', 'on');
3893                                    end
3894                                end
3895                            end
3896
3897                            %---------------------------------
3898                            %------Arrows: Plot Vector-------
3899                            %---------------------------------
3900                            if app.PlotvectorsCheckBox.Value
3901                                if any(~contains(app.FileNAME5, "umax") | app.↙
VorticityButton.Value == false)
3902                                    %------Flip the Matrix-------
3903                                    % Flip horizontally
```

```
3904                                if app.HorizontalCheckBox_2.Value
3905                                    UvectorArrow =  fliplr(Uvector);
3906                                    VvectorArrow =  fliplr(Vvector);
3907                                else
3908                                    UvectorArrow =  Uvector;
3909                                    VvectorArrow =  Vvector;
3910                                end
3911
3912                                % Flip vertically
3913                                if app.VerticalCheckBox_2.Value
3914                                    UvectorArrow =  flipud(Uvector);
3915                                    VvectorArrow =  flipud(Vvector);
3916                                else
3917                                    UvectorArrow =  Uvector;
3918                                    VvectorArrow =  Vvector;
3919                                end
3920
3921                                q1 = quiver(gca(i),app.xcoord(1:app.↙
ArrowSpacing.Value:end,1:app.ArrowSpacing.Value:end),app.ycoord(1:app.ArrowSpacing.↙
Value:end,1:app.ArrowSpacing.Value:end),UvectorArrow(1:app.ArrowSpacing.Value:end,↙
1:app.ArrowSpacing.Value:end),VvectorArrow(1:app.ArrowSpacing.Value:end,1:app.↙
ArrowSpacing.Value:end),app.arrowscale.Value);
3922                                q1.Color = quivercolor;
3923                                q1.LineWidth = app.ArrowThickness.Value;
3924                                q1.MaxHeadSize = app.ArrowHeadSize.Value;
3925                                axis equal  %use equal unit lengths
3926                            end
3927                        end
3928
3929                        %-----------------------------------------
3930                        %------GENERAL LAYOUT: Boundary limit-------
3931                        %-----------------------------------------
3932                        xlim([xMin_VM,xMax_VM])
3933                        ylim([yMin_VM,yMax_VM])
3934                        hold on
3935                        f.Visible = 'on';
3936
3937                        %-----------------------------------
3938                        %------GENERAL LAYOUT: Ticks-------
3939                        %-----------------------------------
3940                        %Reduce the number of ticks to 2 on the X axis and 2↙
on the Y axis for simplicity
3941                        % Get the current ticks from the first plot
3942                        if i == 1
3943                            xAllTicks = xticks(gca);
3944                            yAllTicks = yticks(gca);
3945
3946                            % Ensure the ticks are numeric arrays
3947                            if iscell(xAllTicks)
3948                                xAllTicks = cellfun(@str2double, xAllTicks,↙
'UniformOutput', true);
3949                            end
3950                            if iscell(yAllTicks)
3951                                yAllTicks = cellfun(@str2double, yAllTicks,↙
'UniformOutput', true);
3952                            end
3953
3954                            % Remove any NaN values that might result from↙
```

```
str2double conversion
3955                                xAllTicks = xAllTicks(~isnan(xAllTicks));
3956                                yAllTicks = yAllTicks(~isnan(yAllTicks));
3957
3958                                % Select only the 2nd and second-to-last ticks
3959                                % For X axis
3960                                if length(xAllTicks) >= 4  % Check first if there↙
are enough ticks to select from
3961                                    xticksSelected = sort([xAllTicks(2), xAllTicks↙
(end-1)]);
3962                                elseif length(xAllTicks) >= 2
3963                                    xticksSelected = sort([xAllTicks(1), xAllTicks↙
(end)]); % Use the first and last if less than 4
3964                                else
3965                                    xticksSelected = xAllTicks; % Use whatever↙
ticks are available
3966                                end
3967
3968                                % For Y axis
3969                                if length(yAllTicks) >= 4
3970                                    yticksSelected = sort([yAllTicks(2), yAllTicks↙
(end-1)]);
3971                                elseif length(yAllTicks) >= 2
3972                                    yticksSelected = sort([yAllTicks(1), yAllTicks↙
(end)]); % Use the first and last if less than 4
3973                                else
3974                                    yticksSelected = yAllTicks; % Use whatever↙
ticks are available
3975                                end
3976
3977                            end
3978
3979                            % Update the number of ticks on the axes
3980                            xticks(gca, xticksSelected);
3981                            yticks(gca, yticksSelected);
3982
3983                            % Determine the maximum decimal places for x and y↙
ticks
3984                            xmaxDecimals = max(cellfun(@(x) length(regexp(x, '(?↙
<=\.)\d+', 'match', 'once')), ...
3985                                cellstr(num2str(xticksSelected'))));
3986                            ymaxDecimals = max(cellfun(@(y) length(regexp(y, '(?↙
<=\.)\d+', 'match', 'once')), ...
3987                                cellstr(num2str(yticksSelected'))));
3988
3989                            % Convert tick labels to their equivalent in degrees
3990                            if app.PlotindegreesCheckBox.Value
3991                                % Convert selected ticks to formatted strings and↙
set them for degrees
3992                                xticklabels(gca, arrayfun(@(x) sprintf('%.1f°',↙
mod(x + 360, 360)), xticksSelected, 'UniformOutput', false));
3993                                yticklabels(gca, arrayfun(@(y) sprintf('%.1f°',↙
y), yticksSelected, 'UniformOutput', false));
3994                            else
3995                                % Correct the format string and apply calculated↙
maximum decimals
3996                                xticklabels(gca, arrayfun(@(x) sprintf(['%.'↙
num2str(xmaxDecimals) 'f'], x), xticksSelected, 'UniformOutput', false));
```

```
3997                            yticklabels(gca, arrayfun(@(y) sprintf(['%.'↙
num2str(ymaxDecimals) 'f'], y), yticksSelected, 'UniformOutput', false));
3998                        end
3999
4000                        % Set the y-axis tick angle
4001                        ytickangle(gca, -270);
4002
4003                        %Fontsize of the tick labels
4004                        set(gca,'Layer', 'top','LineWidth',1,'FontSize',app.↙
ColorbarTextSize.Value-1)
4005
4006                        %Adjust tick lengths based on the number of subplots↙
if 'Plot all data in one figure' is selected
4007                        if FileLength1 <= 2 && app.↙
Plotalldatainonefigure Button_2.Value
4008                            set(gca,'TickLength',[0.015 0.015])
4009                        else
4010                            set(gca,'TickLength',[0.01 0.01])
4011                        end
4012
4013                        grid off
4014                        box on
4015                        daspect([1 1 1])
4016
4017                        %--------------------------------------
4018                        %------GENERAL LAYOUT: Map Title------
4019                        %--------------------------------------
4020                        str = string(app.FileNAME5(i));
4021
4022                        dbl1 = extract(str,pat); % Extract numbers from the↙
filename to calculate the time frame
4023
4024                        if startTime == 0 && str2double(dbl1) == 0 %Set the↙
time to zero
4025                              timeT = startTime + (str2double(dbl1)-1);
4026                        else
4027                            OriginaltimeT2 = totalSimulationTime * (str2double↙
(dbl1)-1); % Convert the filenumber to time
4028                            timeT = startTime + OriginaltimeT2; % Adjust the↙
start of the time count
4029                        end
4030
4031
4032                        % Time conversion
4033                        if timeT < 60
4034                            timeStr = sprintf("%d sec", timeT);
4035                        elseif timeT < 3600
4036                            minutes = round(timeT / 60);
4037                            if minutes == 1
4038                                timeStr = "1 min";
4039                            else
4040                                timeStr = sprintf("%d mins", minutes);
4041                            end
4042                        else
4043                            raw_hours = timeT/3600;
4044                            hours = fix(timeT/3600); %Extract the whole↙
number%Extract the whole number
4045                            hours decimal = raw hours - floor(raw hours); %↙
```

```
Extract the decimal places
                            minutes = round(hours_decimal *  60);

                            if hours > 0
                                if minutes > 0
                                    timeStr = sprintf("%d hr, %d mins", hours,↙
minutes);
                                else
                                    timeStr = sprintf("%d hr", hours);
                                end
                            end
                        end

                        title(timeStr, 'FontSize',colorbarTextSize); %Add the↙
title to the top of the map; Adjust the font size of the title
                        hold on


                        %---------------------------------------
                        %------GENERAL LAYOUT: Figure Size-------
                        %---------------------------------------
                        %Figure Size
                        if ~app.AutoSetCheckBox_3.Value
                            figureHandle = ancestor(p, 'figure'); % Get the↙
figure handle containing the axes
                            set(figureHandle, 'Units', 'Inches', 'Position',↙
[0, 0, app.Width_3.Value, app.Height_3.Value], 'PaperUnits', 'Inches', 'PaperSize',↙
[app.Width_3.Value, app.Height_3.Value]);
% %                             set(gcf, 'Units', 'Inches', 'Position', [0, 0,↙
app.Width_3.Value, app.Height_3.Value], 'PaperUnits', 'Inches', 'PaperSize', [app.↙
Width_3.Value, app.Height_3.Value]);
                        end

                        %Create animation / closing the file
                        if app.mp4CheckBox_2.Value
                            set(gcf,'Renderer','zbuffer');
                            f = getframe(gcf);
                            writeVideo(vidObj,f);
                        end

                        %---------------------------------------
                        %------SAVE MAP: Plot Separately-------
                        %---------------------------------------
                        if app.PlotseparatelyButton_3.Value
                            %Save the maps with filenames matching the loaded↙
file
                            OF  = fullfile(app.FigureDirectoryVector,app.↙
Ufiles(i));


                            % Set the figure size for exporting
                            if ~app.AutoSetCheckBox_3.Value
                                set(figureHandle, 'Units', 'Inches',↙
'Position', [0, 0, app.Width_3.Value, app.Height_3.Value], ...
                                    'PaperUnits', 'Inches', 'PaperSize', [app.↙
Width_3.Value, app.Height_3.Value]);
                                set(gcf, 'Units', 'Inches', 'Position', [0, 0,↙
app.Width_3.Value, app.Height_3.Value], ...
```

```
4091                                    'PaperUnits', 'Inches', 'PaperSize', [app.↙
Width_3.Value, app.Height_3.Value]);
4092                                end
4093
4094                                %Options for saving format
4095                                %PNG
4096                                if app.pngCheckBox_3.Value
4097                                    outputfile = OF + ".png";
4098                                    exportgraphics(gcf,outputfile,'Resolution',↙
300)
4099                                end
4100
4101                                %JPG
4102                                if app.jpgCheckBox_3.Value
4103                                    outputfile = OF + ".jpg";
4104                                    exportgraphics(gcf,outputfile,'Resolution',↙
300)
4105                                end
4106
4107                                %PDF
4108 %                                if app.pdfCheckBox_3.Value
4109 %                                    outputfile = OF + ".pdf";
4110 %                                    exportgraphics(gcf,↙
outputfile,'ContentType','vector','Resolution',300)
4111 %                                end
4112
4113                                %TXT - to save the raw vector files
4114                                if app.txtCheckBox_2.Value
4115                                    outputfile1 = OF + "_U.txt";
4116                                    outputfile2 = OF + "_V.txt";
4117                                    writematrix(Uvector,outputfile1);
4118                                    dlmwrite(outputfile1,Uvector,'delimiter',↙
'\t');
4119                                    writematrix(Vvector,outputfile2);
4120                                    dlmwrite(outputfile2,Vvector,'delimiter',↙
'\t');
4121                                end
4122
4123                                %TIF
4124                                if app.tifCheckBox.Value
4125                                    XXX = app.xcoord;
4126                                    YYY = app.ycoord;
4127
4128                                    west_limit = app.WestEditField_2.Value;
4129                                    east_limit = app.EastEditField_2.Value;
4130                                    north_limit = app.NorthEditField_2.Value;
4131                                    south_limit = app.SouthEditField_2.Value;
4132
4133                                    %Make the longitude values within -180 to 180↙
range
4134                                    if west_limit ~= 180
4135                                        west_limit = mod(west_limit + 180, 360) -↙
180;
4136                                    end
4137
4138                                    if east_limit ~= 180
4139                                        east_limit = mod(east_limit + 180, 360) -↙
180;
```

```
4140                                    end
4141
4142                                    % Convert x coordinates to correct degree↙
range
4143                                    XXX(XXX > 180) = XXX(XXX > 180) - 360;
4144
4145                                    % Create meshgrid
4146                                    [X, Y] = meshgrid(XXX, YYY);
4147                                    Z = VMdata;
4148
4149                                    %Check if the latitude is within the↙
Geographic limits
4150                                    if app.LatitudeEditField_2.Value > 90 || app.↙
LatitudeEditField_2.Value < -90
4151                                        app.LatitudeEditField_2.Value = 0;
4152                                    else
4153                                        %Clip the matrix based on boundary limits
4154                                        if west_limit >= 0 && east_limit >=0  ||↙
west_limit < 0 && east_limit < 0 % if east and west values are either both (+) or↙
both (-) values
4155                                            x_indices_positive =  XXX >=↙
west_limit & XXX <= east_limit;
4156                                            y_indices = YYY >= south_limit & YYY↙
<= north_limit;
4157                                            Z_positive = Z(y_indices,↙
x_indices_positive);
4158                                            X_positive = X(y_indices,↙
x_indices_positive);
4159                                            Y = Y(y_indices, x_indices_positive);↙
% Keep the same for both file
4160                                            if ~isempty(Z_positive)
4161                                                % Define spatial referencing↙
information
4162                                                R_positive = georasterref↙
('RasterSize', size(Z_positive), 'LatitudeLimits', [min(Y(:)), max(Y(:))],↙
'LongitudeLimits', [min(X_positive(:)), max(X_positive(:))]);
4163                                                R_positive.ColumnsStartFrom =↙
'south'; % Set the column orientation to start from the north
4164                                                R_positive.RowsStartFrom =  'west';
4165
4166                                                % Save as geotiff
4167                                                filename_positive = OF +  ".tif";
4168                                                geotiffwrite(filename_positive,↙
Z_positive, R_positive);
4169                                            end
4170
4171                                        elseif west_limit >= 0 && east_limit < 0↙
%if longitude values are a combination of (+) west limit and (-) east limit
4172                                            x_indices_positive = XXX >= 0 & XXX >=↙
west_limit;
4173                                            x_indices_negative = XXX < 0 & XXX <=↙
east_limit;
4174                                            y_indices = YYY >= south_limit & YYY↙
<= north_limit;
4175                                            Z_positive = Z(y_indices,↙
x_indices_positive);
4176                                            Z_negative = Z(y_indices,↙
x_indices_negative);
```

```
4177                                            X_positive = X(y_indices, ↙
x_indices_positive);
4178                                            X_negative = X(y_indices, ↙
x_indices_negative);
4179                                            Y = Y(y_indices, x_indices_positive); ↙
% Keep the same for both file
4180
4181                                            if ~isempty(Z_positive)
4182                                                % Define spatial referencing↙
information
4183                                                R_positive = georasterref ↙
('RasterSize', size(Z_positive), 'LatitudeLimits', [min(Y(:)), max(Y(:))], ↙
'LongitudeLimits', [min(X_positive(:)), max(X_positive(:))]);
4184                                                R_positive.ColumnsStartFrom = ↙
'south'; % Set the column orientation to start from the north
4185                                                R_positive.RowsStartFrom = 'west';
4186
4187                                                % Save as geotiff
4188                                                filename_positive = OF + "_1.tif";
4189                                                geotiffwrite(filename_positive, ↙
Z_positive, R_positive);
4190                                            end
4191
4192                                            if ~isempty(Z_negative)
4193                                                % Define spatial referencing↙
information
4194                                                R_negative = georasterref ↙
('RasterSize', size(Z_negative), 'LatitudeLimits', [min(Y(:)), max(Y(:))], ↙
'LongitudeLimits', [min(X_negative(:)), max(X_negative(:))]);
4195                                                R_negative.ColumnsStartFrom = ↙
'south'; % Set the column orientation to start from the north
4196                                                R_negative.RowsStartFrom = 'west';
4197
4198                                                % Save as geotiff
4199                                                filename_negative = OF + "_2.tif";
4200                                                geotiffwrite(filename_negative, ↙
Z_negative, R_negative)
4201                                            end
4202
4203                                        elseif west_limit < 0 && east_limit >= 0↙
%if longitude values are a combination of (-) west limit and (+) east limit
4204                                            x_indices_positive = XXX >= 0 & XXX <= ↙
east_limit;
4205                                            x_indices_negative = XXX < 0 & XXX >= ↙
west_limit;
4206                                            y_indices = YYY >= south_limit & YYY ↙
<= north_limit;
4207                                            Z_positive = Z(y_indices, ↙
x_indices_positive);
4208                                            Z_negative = Z(y_indices, ↙
x_indices_negative);
4209                                            X_positive = X(y_indices, ↙
x_indices_positive);
4210                                            X_negative = X(y_indices, ↙
x_indices_negative);
4211                                            Y = Y(y_indices, x_indices_positive); ↙
% Keep the same for both file
4212
```

```
4213                                            if ~isempty(Z_positive)
4214                                                % Define spatial referencing↙
information
4215                                                R_positive = georasterref↙
('RasterSize', size(Z_positive), 'LatitudeLimits', [min(Y(:)), max(Y(:))],↙
'LongitudeLimits', [min(X_positive(:)), max(X_positive(:))]);
4216                                                R_positive.ColumnsStartFrom =↙
'south'; % Set the column orientation to start from the north
4217                                                R_positive.RowsStartFrom = 'west';
4218
4219                                                % Save as geotiff
4220                                                filename_positive = OF + "_1.tif";
4221                                                geotiffwrite(filename_positive,↙
Z_positive, R_positive);
4222                                            end
4223                                            if ~isempty(Z_negative)
4224                                                % Define spatial referencing↙
information
4225                                                R_negative = georasterref↙
('RasterSize', size(Z_negative), 'LatitudeLimits', [min(Y(:)), max(Y(:))],↙
'LongitudeLimits', [min(X_negative(:)), max(X_negative(:))]);
4226                                                R_negative.ColumnsStartFrom =↙
'south'; % Set the column orientation to start from the north
4227                                                R_negative.RowsStartFrom = 'west';
4228
4229                                                % Save as geotiff
4230                                                filename_negative = OF + "_2.tif";
4231                                                geotiffwrite(filename_negative,↙
Z_negative, R_negative);
4232                                            end
4233                                        end
4234                                    end
4235                                end
4236                            end
4237                        end
4238                    end
4239
4240                    %Change colorbar size
4241                            if cb.Label.FontSize ~= colorbarTextSize
4242                                cb.Label.FontSize =colorbarTextSize;
4243                            end
4244
4245                    %-------------------------------------------------
4246                    %------SAVE MAP: Plot all data in one figure-------
4247                    %-------------------------------------------------
4248                    if app.PlotalldatainonefigureButton_2.Value
4249
4250                        % Set the figure size for exporting
4251                        if ~app.AutoSetCheckBox_3.Value
4252                            set(gcf, 'Units', 'Inches', 'Position', [0, 0, app.↙
Width_3.Value, app.Height_3.Value], ...
4253                                'PaperUnits', 'Inches', 'PaperSize', [app.Width_3.↙
Value, app.Height_3.Value]);
4254                        end
4255
4256                        if length(app.Ufiles) == 1
4257                            OF  = fullfile(app.FigureDirectoryVector,app.Ufiles);
4258                        else
```

```
4259                        OF  = fullfile(app.FigureDirectoryVector, ↙
"Vector_Output");
4260                    end
4261
4262                    %PNG
4263                    if app.pngCheckBox_3.Value
4264                        outputfile = OF + ".png";
4265                        exportgraphics(gcf,outputfile,'Resolution',300)
4266                    end
4267
4268                    %JPG
4269                    if app.jpgCheckBox_3.Value
4270                        outputfile = OF + ".jpg";
4271                        exportgraphics(gcf,outputfile,'Resolution',300)
4272                    end
4273
4274                    %PDF
4275 %                    if app.pdfCheckBox_3.Value
4276 %                        outputfile = OF + ".pdf";
4277 %                        exportgraphics(gcf, ↙
outputfile,'ContentType','vector','Resolution',300)
4278 %                    end
4279                end
4280
4281        end
4282
4283        % Value changed function: QuiverColorDropDown
4284        function QuiverColorDropDownValueChanged(app, event)
4285
4286        end
4287
4288        % Button pushed function: CloseFiguresButton_3
4289        function CloseFiguresButton_3Pushed(app, event)
4290            close all
4291        end
4292
4293        % Button pushed function: Button_22
4294        function Button_22Pushed(app, event)
4295            clear  app.bathymetryinputdata2 app.bathymetrydata2
4296
4297            %Load the bathymetry file and display its filename in the textbox
4298            [bathymetryfile,path2] = uigetfile('*'); %*.txt;*.out;*.tif'
4299            if ~(ismcc || isdeployed)
4300                addpath(genpath(fullfile(string(path2))));
4301            end
4302            app.DepthFileEditField_2.Value = string(bathymetryfile);
4303            app.bathymetryinputdata2 = fullfile(string(path2), string ↙
(bathymetryfile));
4304            app.bathymetryname2 = string(bathymetryfile);
4305            app.DepthFileEditField_2.BackgroundColor = 'w';
4306            app.DepthFileEditField_2.FontColor =  'k';
4307
4308
4309            if app.DepthFileEditField_2.Value == string(bathymetryfile)
4310                if ~strcmp(app.DepthFileEditField_2.Value, "0") && ~strcmp ↙
(app.DepthFileEditField_2.Value, "00")
4311                    try
4312                        if contains(app.bathymetryinputdata2,'.tif')
```

```
4313                            [A1,R2] = readgeoraster(app.bathymetryinputdata2);
4314                            app.bathymetrydata2 =  flipud(A1);
4315
4316                            %Extract lat and long of the southwest corner
4317                            app.LongitudeEditField_2.Value = min(R2.↙
LongitudeLimits);
4318                            app.LatitudeEditField_2.Value = min(R2.↙
LatitudeLimits);
4319                            app.LongitudeEditField_2.FontColor = 'k';
4320                            app.LatitudeEditField_2.FontColor = 'k';
4321
4322                            %Extract grid size
4323                            app.gridX_2.Value = R2.CellExtentInLatitude;
4324                            app.gridY_2.Value = R2.CellExtentInLongitude;
4325                            app.gridX_2.FontColor = 'k';
4326                            app.gridY_2.FontColor = 'k';
4327
4328                            %Update the X-axis limits in the 'General Layout'↙
section
4329                            app.xcoord = [0:app.col1vel] * app.gridX_2.Value+↙
app.LongitudeEditField_2.Value;
4330                            app.EastEditField_2.Value = max(app.xcoord);
4331                            app.WestEditField_2.Value = min(app.xcoord);
4332
4333                            %Update the Y-axis limits in the 'General Layout'↙
section
4334                            app.ycoord = [0:app.row1vel] * app.gridY_2.Value+↙
app.LatitudeEditField_2.Value;
4335                            app.NorthEditField_2.Value = max(app.ycoord);
4336                            app.SouthEditField_2.Value = min(app.ycoord);
4337
4338                            app.EastEditField_2.FontColor = 'k';
4339                            app.WestEditField_2.FontColor = 'k';
4340                            app.NorthEditField_2.FontColor = 'k';
4341                            app.SouthEditField_2.FontColor = 'k';
4342                            drawnow
4343
4344                            if app.BathymetryButton.Value %If bathymetry is↙
selected as the basemap
4345                                app.FileTextArea_5.Value = app.↙
DepthFileEditField_2.Value; %show the bathymetry filename in the textbox in↙
'Basemap and Overlays' section
4346                            else
4347                                app.FileTextArea_5.Value = '';
4348                            end
4349                            drawnow
4350
4351                        else %Only read the matrix for use as a basemap and↙
for plotting bathymetry contours
4352                            app.bathymetrydata2 = readmatrix(app.↙
bathymetryinputdata2);
4353                        end
4354
4355                    end
4356                    %Enable the 'Land Color' dropdown list & the Plot↙
Bathymetry Contours option in 'Basemap and Overlays' section
4357                    app.LandColor.Enable = "on";
4358                    app.PlotBathymetryContoursCheckBox.Enable = "on";
```

```
4359                 end
4360             end
4361
4362             %Check if the uploaded file has the required file format to log error
4363             if ~contains(bathymetryfile, '.txt') && ~contains(bathymetryfile, '.tif') && ~contains(bathymetryfile, 'mask_')
4364                 app.bathymetrycheck = 1;
4365             end
4366
4367         end
4368
4369         % Callback function
4370         function ButtonGroup_13SelectionChanged(app, event)
4371
4372
4373
4374         end
4375
4376         % Value changed function: PlotindegreesCheckBox_2
4377         function PlotindegreesCheckBox_2ValueChanged(app, event)
4378
4379
4380         end
4381
4382         % Callback function
4383         function Button_23Pushed(app, event)
4384
4385
4386
4387         end
4388
4389         % Value changed function: FileTextArea_5
4390         function FileTextArea_5ValueChanged(app, event)
4391
4392
4393         end
4394
4395         % Selection changed function: ButtonGroup_12
4396         function ButtonGroup_12SelectionChanged(app, event)
4397             if app.PlotalldatainonefigureButton_2.Value
4398                 app.mp4CheckBox_2.Enable = "off";
4399                 app.FramerateEditFieldLabel.Visible = "off";
4400                 app.FramerateEditField.Visible = "off";
4401                 app.txtCheckBox_2.Enable  = "off";
4402                 app.tifCheckBox.Enable = "off";
4403             elseif app.PlotseparatelyButton_3.Value
4404                 app.mp4CheckBox_2.Enable = "on";
4405                 app.txtCheckBox_2.Enable  = "on";
4406                 app.tifCheckBox.Enable = "on";
4407                 if  app.mp4CheckBox_2.Value %If it is left checked, show the frame rate option
4408                     app.FramerateEditFieldLabel.Visible = "on";
4409                     app.FramerateEditField.Visible = "on";
4410                 else
4411                     app.FramerateEditFieldLabel.Visible = "off";
4412                     app.FramerateEditField.Visible = "off";
4413                 end
```

```
4414            end
4415        end
4416
4417        % Callback function
4418        function CurrentDirectionsCheckBoxValueChanged(app, event)
4419
4420        end
4421
4422        % Callback function
4423        function Button_24Pushed(app, event)
4424            %LOAD THE U VECTOR FILES
4425            [files, path] = uigetfile('*.*','Select the files','Code↙
files','MultiSelect', 'on');
4426
4427            %Collect all the files
4428            allfiles = string(path) + string(files);
4429            app.Uvectorfullfile = sort(allfiles);
4430            FileNAME = sort(string(files));
4431            app.FileTextArea_6.Value = string(FileNAME); %display the filename↙
list
4432            app.FileTextArea_6.FontColor = 'k';
4433            app.FileTextArea_6.BackgroundColor = 'w';
4434
4435            %Find the equivalent V Vector files
4436            vfiles = string(files);
4437            vfilenames = regexprep(vfiles,"u","v",'once');  %replaces only the↙
first letter "u" in each of the filenames
4438            app.Vvectorfullfile = string(path) + vfilenames;
4439            app.FileTextArea_7.Value = vfilenames;
4440            app.FileTextArea_7.FontColor = 'k';
4441            app.FileTextArea_7.BackgroundColor = 'w';
4442
4443
4444            %Check if any file is loaded
4445            if matches(app.FileTextArea_3.Value,"0") || matches(app.↙
FileTextArea_3.Value,"00")
4446                %do nothing
4447            else
4448                %READ THE FIRST U FILE TO SETUP MATRIX SIZE AND MAP BOUNDARY↙
LIMIT
4449                f = readmatrix(app.Ufile(1));
4450                %SETUP OF THE COORDINATES
4451                [row,col] = size(f);
4452                col1 = col-1;
4453                row1 = row-1;
4454                app.xcoord = (0:col1) * app.gridX_2.Value+ app.↙
LongitudeEditField_2.Value;
4455                app.ycoord = [0:row1] * app.gridY_2.Value+ app.↙
LatitudeEditField_2.Value;
4456
4457                %SET UP THE MAP BOUNDARY
4458                app.EastEditField_2.Value = max(app.xcoord);
4459                app.WestEditField_2.Value = min(app.xcoord);
4460                app.NorthEditField_2.Value = max(app.ycoord);
4461                app.SouthEditField_2.Value = min(app.ycoord);
4462            end
4463
4464        end
```

```
4465
4466        % Value changed function: FileTextArea_3
4467        function FileTextArea_3ValueChanged(app, event)
4468
4469
4470        end
4471
4472        % Size changed function: VelocityMapTab
4473        function VelocityMapTabSizeChanged(app, event)
4474
4475
4476        end
4477
4478        % Callback function
4479        function PlotColormapCheckBoxValueChanged(app, event)
4480            if app.PlotColormapCheckBox.Value
4481                app.BackgroundMapColorDropDown.Enable = "on";
4482                app.FlipCheckBox_4.Enable = "on";
4483                app.InterpolationDivisionEditField_2.Enable = "on";
4484                app.MaxBarValue.Enable = "on";
4485                app.MinBarValue.Enable = "on";
4486            else
4487                app.BackgroundMapColorDropDown.Enable = "off";
4488                app.FlipCheckBox_4.Enable = "off";
4489                app.InterpolationDivisionEditField_2.Enable = "off";
4490                app.MaxBarValue.Enable = "off";
4491                app.MinBarValue.Enable = "off";
4492            end
4493
4494            if app.PlotColormapCheckBox.Value && app.etaButton.Value
4495                app.Button_23.Enable = "on";
4496            else
4497                app.Button_23.Enable = "off";
4498            end
4499
4500            if app.BathymetryButton.Value
4501                app.Button_23.Enable = "off";
4502            end
4503        end
4504
4505        % Value changed function: PlotBathymetryContoursCheckBox
4506        function PlotBathymetryContoursCheckBoxValueChanged(app, event)
4507            if app.PlotBathymetryContoursCheckBox.Value
4508                app.WidthEditField_2.Enable = "on";
4509                app.StyleDropDown_3.Enable = "on";
4510                app.IntervalEditField_6.Enable = "on";
4511                app.MaximumEditField_2.Enable = "on";
4512                app.MinimumEditField_2.Enable = "on";
4513                app.ColorDropDown_4.Enable = "on";
4514                app.AddLabelCheckBox_2.Enable = "on";
4515                if app.AddLabelCheckBox_2.Value
4516                    app.IntervalEditField_8.Enable = "on";
4517                    app.LabelSizeCont.Enable = "on";
4518                    app.SpacingEditField.Enable = "on";
4519                else
4520                    app.IntervalEditField_8.Enable = "off";
4521                    app.LabelSizeCont.Enable = "off";
4522                    app.SpacingEditField.Enable = "off";
```

```
4523                end
4524            else
4525                app.IntervalEditField_8.Enable = "off";
4526                app.WidthEditField_2.Enable = "off";
4527                app.StyleDropDown_3.Enable = "off";
4528                app.IntervalEditField_6.Enable = "off";
4529                app.MaximumEditField_2.Enable = "off";
4530                app.MinimumEditField_2.Enable = "off";
4531                app.ColorDropDown_4.Enable = "off";
4532                app.AddLabelCheckBox_2.Enable = "off";
4533                app.LabelSizeCont.Enable = "off";
4534                app.SpacingEditField.Enable = "off";
4535            end
4536        end
4537
4538        % Size changed function: Panel_16
4539        function Panel_16SizeChanged(app, event)
4540
4541
4542        end
4543
4544        % Size changed function: BasemapTab
4545        function BasemapTabSizeChanged(app, event)
4546
4547        end
4548
4549        % Value changed function: DepthFileEditField_2
4550        function DepthFileEditField_2ValueChanged(app, event)
4551
4552        end
4553
4554        % Value changed function: AutoSetCheckBox_3
4555        function AutoSetCheckBox_3ValueChanged(app, event)
4556            if app.AutoSetCheckBox_3.Value
4557                app.Height_3.Enable = "off";
4558                app.Width_3.Enable = "off";
4559            else
4560                app.Height_3.Enable = "on";
4561                app.Width_3.Enable = "on";
4562            end
4563        end
4564
4565        % Size changed function: ButtonGroup_12
4566        function ButtonGroup_12SizeChanged(app, event)
4567
4568        end
4569
4570        % Button pushed function: Button_26
4571        function Button_26Pushed(app, event)
4572            workingfolder = uigetdir;
4573            Dir1 = fullfile(string(workingfolder), 'OUTPUT_FILES');
4574            app.WORKFOLDER3 = Dir1;
4575            FigureFolder = fullfile(Dir1, 'Figures');
4576            app.OutputDirectoryEditField_3.Value = FigureFolder;
4577            app.WORKFOLDER4 = fullfile(deblank(FigureFolder));
4578            app.OutputDirectoryEditField_3.FontColor = 'k';
4579
4580            %Create the folders
```

```
4581            if ~exist(Dir1, 'dir')
4582                mkdir(Dir1);
4583            end
4584            if ~exist(FigureFolder, 'dir')
4585                mkdir(FigureFolder);
4586            end
4587        end
4588
4589        % Value changed function: CoastlinecolourDropDown_7
4590        function CoastlinecolourDropDown_7ValueChanged(app, event)
4591            value = app.CoastlinecolourDropDown_7.Value;
4592            %List of values based on the cbrewer2 colormap
4593            validValues = {'blue', 'blue - green', 'blue - purple', 'green -↙
blue', ...
4594                'greens', 'grays', 'oranges', 'orange - red', ...
4595                'purple - blue', 'purple - blue - green', 'purple - red', ...
4596                'purples', 'red - purple', 'reds', 'yellow - green', ...
4597                'yellow - green - blue', 'yellow - orange - brown', ...
4598                'yellow - orange - red', 'brown - teal', 'pink - light green',↙
...
4599                'purple - green', 'purple - orange', 'red - blue', ...
4600                'red - gray', 'red - yellow - blue', 'red - yellow - green',↙
...
4601                'spectral', 'accent', 'dark 2', 'paired', 'pastel 1', ...
4602                'pastel 2', 'set 1', 'set 2', 'set 3'};
4603
4604            % If the selected value in the dropdown list is part of the↙
cbrewer2 colormap, enable the color interpolation textbox
4605            if ismember(value, validValues)
4606                app.ColorinterpolationEditField_2.Enable = "on";
4607                app.DivisionLabel.Enable = "on";
4608            else
4609                app.ColorinterpolationEditField_2.Enable = "off";
4610                app.DivisionLabel.Enable = "off";
4611            end
4612        end
4613
4614        % Value changed function: BackgroundMapColorDropDown
4615        function BackgroundMapColorDropDownValueChanged(app, event)
4616            cbrewercolors = {'blue', 'blue - green', 'blue - purple', 'green -↙
blue', 'greens', ...
4617                'grays', 'oranges', 'orange - red', 'purple - blue', 'purple -↙
blue - green', ...
4618                'purple - red', 'purples', 'red - purple', 'reds', 'yellow -↙
green', ...
4619                'yellow - green - blue', 'yellow - orange - brown', 'yellow -↙
orange - red', ...
4620                'brown - teal', 'pink - light green', 'purple - green',↙
'purple - orange', ...
4621                'red - blue', 'red - gray', 'red - yellow - blue', 'red -↙
yellow - green', ...
4622                'spectral', 'accent', 'dark 2', 'paired', 'pastel 1', 'pastel↙
2', ...
4623                'set 1', 'set 2', 'set 3'};
4624
4625            %Activate the interpolation division when the selected colormap is↙
of cbrewer type
4626            if ismember(app.BackgroundMapColorDropDown.Value, cbrewercolors)
```

```
4627                 app.InterpolationDivisionEditField_2.Enable = "on";
4628                 app.InterpolationDivisionEditFieldLabel_2.Enable = "on";
4629             else
4630                 app.InterpolationDivisionEditField_2.Enable = "off";
4631                 app.InterpolationDivisionEditFieldLabel_2.Enable = "off";
4632             end
4633
4634
4635
4636         end
4637
4638         % Drop down opening function: BackgroundMapColorDropDown
4639         function BackgroundMapColorDropDownOpening(app, event)
4640             app.IntervalEditField_6.FontColor = 'k';
4641         end
4642
4643         % Value changed function: IntervalEditField_6
4644         function IntervalEditField_6ValueChanged(app, event)
4645             app.IntervalEditField_6.FontColor = 'k';
4646             app.IntervalEditField_8.Value = app.IntervalEditField_6.Value;
4647         end
4648
4649         % Button pushed function: SettoDefaultButton
4650         function SettoDefaultButtonPushed(app, event)
4651             %Restore the boundary extent using input from the 'Input Data' ↙
section
4652
4653             %Extract row and column dimensions from the first imported file
4654             f = readmatrix(app.Uvectorfullfile(1));
4655
4656             %Determine the coordinates
4657             [row,col] = size(f);
4658             col1 = col-1;
4659             row1 = row-1;
4660             app.xcoord = [0:col1] * app.gridX_2.Value+ app. ↙
LongitudeEditField_2.Value;
4661             app.ycoord = [0:row1] * app.gridY_2.Value+ app. ↙
LatitudeEditField_2.Value;
4662
4663             %Show the map boundary in the textboxes in 'General Layout'
4664             app.EastEditField_2.Value = max(app.xcoord);
4665             app.WestEditField_2.Value = min(app.xcoord);
4666             app.NorthEditField_2.Value = max(app.ycoord);
4667             app.SouthEditField_2.Value = min(app.ycoord);
4668
4669             app.EastEditField_2.FontColor = 'k';
4670             app.WestEditField_2.FontColor = 'k';
4671             app.NorthEditField_2.FontColor = 'k';
4672             app.SouthEditField_2.FontColor = 'k';
4673             drawnow
4674
4675         end
4676
4677         % Button pushed function: SettoDefaultButton_2
4678         function SettoDefaultButton_2Pushed(app, event)
4679             %Restore the boundary extent using input from the 'Input Data' ↙
section
4680
```

```
4681            %Extract row and column dimensions from the first imported file
4682            f = readmatrix(app.FileINPUT(1));
4683            [row,col] = size(f);
4684
4685            %Determine the coordinates
4686            col1 = col-1;
4687            row1 = row-1;
4688            app.x = app.LongitudeEditField.Value + [0:col1] * app.gridX.Value;
4689            app.y = app.LatitudeEditField.Value + [0:row1] * app.gridY.Value;
4690
4691
4692            %Show the map boundary in the textboxes in 'General Layout'
4693            app.EastEditField.Value = max(app.x);
4694            app.WestEditField.Value = min(app.x);
4695            app.NorthEditField.Value = max(app.y);
4696            app.SouthEditField.Value = min(app.y);
4697            drawnow
4698
4699            app.EastEditField.FontColor = 'k';
4700            app.WestEditField.FontColor = 'k';
4701            app.NorthEditField.FontColor = 'k';
4702            app.SouthEditField.FontColor = 'k';
4703
4704        end
4705
4706        % Size changed function: Panel_15
4707        function Panel_15SizeChanged(app, event)
4708
4709
4710        end
4711
4712        % Value changed function: PlotGaugesCheckBox
4713        function PlotGaugesCheckBoxValueChanged(app, event)
4714            if app.PlotGaugesCheckBox.Value
4715                app.FileEditField_2.Enable = "on";
4716                app.Button_27.Enable = "on";
4717                app.ColorDropDown_7.Enable = "on";
4718                app.SizeEditField_2.Enable = "on";
4719                app.gaugemarkerVelocityTab.Enable = "on";
4720                app.AddLabelCheckBox.Enable = "on";
4721                if app.AddLabelCheckBox.Value
4722                    app.GLspacing_2.Enable = "on";
4723                    app.GLspacing_2.Editable = "on";
4724                else
4725                    app.GLspacing_2.Editable = "off";
4726                    app.GLspacingLabel_2.Enable = "off";
4727                end
4728
4729            else
4730                app.FileEditField_2.Enable = "off";
4731                app.Button_27.Enable = "off";
4732                app.ColorDropDown_7.Enable = "off";
4733                app.SizeEditField_2.Enable = "off";
4734                app.gaugemarkerVelocityTab.Enable = "off";
4735                app.AddLabelCheckBox.Enable = "off";
4736            end
4737
4738
```

```
4739        end
4740
4741        % Value changed function: AddLabelCheckBox
4742        function AddLabelCheckBoxValueChanged(app, event)
4743            if app.AddLabelCheckBox.Value
4744                app.FontSizeEditField_2.Enable = "on";
4745                app.FontSizeEditField_2.Editable = "on";
4746                app.CoastlinecolourDropDown_11.Enable = "on";
4747                app.CoastlinecolourDropDown_10.Enable = "on";
4748                app.GLspacing_2.Enable = "on";
4749                app.GLspacing_2.Editable = "on";
4750                app.GLspacingLabel_2.Enable = "on";
4751                app.FontSizeEditField_2Label.Enable = "on";
4752                app.AlignmentLabel_2.Enable = "on";
4753            else
4754                app.FontSizeEditField_2.Enable = "off";
4755                app.FontSizeEditField_2.Editable = "off";
4756                app.CoastlinecolourDropDown_11.Enable = "off";
4757                app.CoastlinecolourDropDown_10.Enable = "off";
4758                app.GLspacing_2.Enable = "off";
4759                app.GLspacingLabel_2.Enable = "off";
4760                app.FontSizeEditField_2Label.Enable = "off";
4761                app.AlignmentLabel_2.Enable = "off";
4762            end
4763
4764        end
4765
4766        % Callback function
4767        function AddLabelsCheckBox_2ValueChanged(app, event)
4768            if app.AddLabelsCheckBox_2.Value
4769                app.IntervalEditField_3.Enable = "on";
4770                app.TextLabelSize.Enable = "on";
4771                app.LabelSpacingEditField.Enable = "on";
4772                app.SpacingEditField_4Label_4.Enable = "on";
4773                app.LabelSpacingEditFieldLabel.Enable = "on";
4774                app.LabelSpacingEditFieldLabel_2.Enable = "on";
4775            else
4776                app.IntervalEditField_3.Enable = "off";
4777                app.TextLabelSize.Enable = "off";
4778                app.LabelSpacingEditField.Enable = "off";
4779                app.SpacingEditField_4Label_4.Enable = "off";
4780                app.LabelSpacingEditFieldLabel.Enable = "off";
4781                app.LabelSpacingEditFieldLabel_2.Enable = "off";
4782            end
4783
4784        end
4785
4786        % Callback function
4787        function AddLabelsCheckBox_3ValueChanged(app, event)
4788            if app.AddLabelsCheckBox_3.Value
4789                app.TextLabelSize_2.Enable = "on";
4790                app.wh_interval.Enable = "on";
4791                app.LabelSpacingEditField_2.Enable = "on";
4792                app.SpacingEditField_4Label_5.Enable = "on";
4793                app.LabelSpacingEditField_2Label_2.Enable = "on";
4794                app.LabelSpacingEditField_2Label.Enable = "on";
4795            else
4796                app.TextLabelSize_2.Enable = "off";
```

```
4797                app.wh_interval.Enable = "off";
4798                app.LabelSpacingEditField_2.Enable = "off";
4799                app.SpacingEditField_4Label_5.Enable =  "off";
4800                app.LabelSpacingEditField_2Label_2.Enable =  "off";
4801                app.LabelSpacingEditField_2Label.Enable =  "off";
4802            end
4803        end
4804
4805        % Callback function
4806        function AddLabelsCheckBox_4ValueChanged(app, event)
4807            if app.AddLabelsCheckBox_4.Value
4808                app.CoastlinecolourDropDown_5.Enable = "on";
4809                app.CoastlinecolourDropDown_4.Enable = "on";
4810                app.SizeEditField_3Label.Enable = "on";
4811                app.AlignmentLabel.Enable = "on";
4812                app.SizeEditField_3.Enable = "on";
4813                app.SizeEditField_3.Editable = "on";
4814                app.GLspacing.Enable = "on";
4815                app.GLspacing.Editable = "on";
4816                app.GLspacingLabel.Enable = "on";
4817                app.FontSizeEditField_2Label.Enable = "on";
4818                app.AlignmentLabel_2.Enable = "on";
4819            else
4820                app.CoastlinecolourDropDown_5.Enable = "off";
4821                app.CoastlinecolourDropDown_4.Enable = "off";
4822                app.SizeEditField_3Label.Enable = "off";
4823                app.AlignmentLabel.Enable = "off";
4824                app.SizeEditField_3.Enable = "off";
4825                app.SizeEditField_3.Editable = "off";
4826                app.GLspacing.Enable = "off";
4827                app.GLspacing.Editable = "off";
4828                app.GLspacingLabel.Enable = "off";
4829                app.FontSizeEditField_2Label.Enable = "off";
4830                app.AlignmentLabel_2.Enable = "off";
4831            end
4832        end
4833
4834        % Size changed function: GaugesTab_2
4835        function GaugesTab_2SizeChanged(app, event)
4836
4837
4838        end
4839
4840        % Button pushed function: Button_27
4841        function Button_27Pushed(app, event)
4842            [filename3,path3] = uigetfile('*.txt;*.shp');
4843            fullname = fullfile(path3,filename3);
4844
4845            [~, name, ext] = fileparts(filename3);
4846            if ~(ismcc || isdeployed)
4847                addpath(genpath(string(path3)));
4848            end
4849
4850            app.GaugeFname3 = name;
4851            app.FileEditField_2.Value = filename3;
4852            app.GAUGEFILE3= filename3;
4853            app.FileEditField_2.FontColor = 'k';
4854            app.FileEditField_2.BackgroundColor = 'w';
```

```
4855
4856            if strcmp(ext, '.shp')
4857                S = shaperead(fullname);
4858                latitudes= [S.Y]';
4859                longitudes= [S.X]';
4860                list = [round(latitudes, 4), round(longitudes, 4)];
4861                list = unique(list, 'rows');
4862                app.latGauge2 = list(:, 1);
4863                app.longGauge2 = list(:, 2);
4864
4865            elseif strcmp(ext, '.txt')
4866                fileID = fopen(fullname, 'r');
4867                dataArray = textscan(fileID, '%f %f', 'Delimiter',↙
'whitespace'); %Read text file and ensure that it is tab delimited
4868                fclose(fileID);
4869                app.latGauge2  = dataArray{1};
4870                app.longGauge2 = dataArray{2};
4871            end
4872
4873            app.FileEditField_2.FontColor = 'k';
4874            app.FileEditField_2.BackgroundColor = 'w';
4875
4876
4877        end
4878
4879        % Value changed function: AddLabelCheckBox_2
4880        function AddLabelCheckBox_2ValueChanged(app, event)
4881            if app.AddLabelCheckBox_2.Value
4882                app.LabelSizeCont.Enable = "on";
4883                app.SpacingEditField.Enable = "on";
4884                app.IntervalEditField_8.Enable = "on";
4885            else
4886                app.LabelSizeCont.Enable = "off";
4887                app.SpacingEditField.Enable = "off";
4888                app.IntervalEditField_8.Enable = "off";
4889            end
4890        end
4891
4892        % Value changed function: LongitudeEditField_2
4893        function LongitudeEditField_2ValueChanged(app, event)
4894            app.LongitudeEditField_2.FontColor = 'k';
4895
4896            %Update the X-axis limits in the 'General Layout' section
4897            app.xcoord = [0:app.col1vel] * app.gridX_2.Value+ app.↙
LongitudeEditField_2.Value;
4898            app.EastEditField_2.Value = max(app.xcoord);
4899            app.WestEditField_2.Value = min(app.xcoord);
4900            drawnow
4901        end
4902
4903        % Value changed function: LatitudeEditField_2
4904        function LatitudeEditField_2ValueChanged(app, event)
4905            app.LatitudeEditField_2.FontColor = 'k';
4906
4907            %Update the Y-axis limits in the 'General Layout' section
4908            app.ycoord = [0:app.row1vel] * app.gridY_2.Value+ app.↙
LatitudeEditField_2.Value;
4909            app.NorthEditField_2.Value = max(app.ycoord);
```

```
4910            app.SouthEditField_2.Value = min(app.ycoord);
4911            drawnow
4912        end
4913
4914        % Value changed function: gridX_2
4915        function gridX_2ValueChanged(app, event)
4916            app.gridX_2.FontColor = 'k';
4917
4918            %Update the X-axis limits in the 'General Layout' section
4919            app.xcoord = [0:app.col1vel] * app.gridX_2.Value+ app.↙
LongitudeEditField_2.Value;
4920            app.EastEditField_2.Value = max(app.xcoord);
4921            app.WestEditField_2.Value = min(app.xcoord);
4922            drawnow
4923        end
4924
4925        % Value changed function: gridY_2
4926        function gridY_2ValueChanged(app, event)
4927            app.gridY_2.FontColor = 'k';
4928
4929            %Update the Y-axis limits in the 'General Layout' section
4930            app.ycoord = [0:app.row1vel] * app.gridY_2.Value+ app.↙
LatitudeEditField_2.Value;
4931            app.NorthEditField_2.Value = max(app.ycoord);
4932            app.SouthEditField_2.Value = min(app.ycoord);
4933            drawnow
4934        end
4935
4936        % Value changed function: TotalSimuilationTimesecEditField_4
4937        function TotalSimuilationTimesecEditField_4ValueChanged(app, event)
4938            app.TotalSimuilationTimesecEditField_4.FontColor = 'k';
4939
4940        end
4941
4942        % Value changed function: arrowscale
4943        function arrowscaleValueChanged(app, event)
4944            app.arrowscale.FontColor = 'k';
4945
4946        end
4947
4948        % Value changed function: ArrowHeadSize
4949        function ArrowHeadSizeValueChanged(app, event)
4950            app.ArrowHeadSize.FontColor = 'k';
4951
4952        end
4953
4954        % Value changed function: ArrowThickness
4955        function ArrowThicknessValueChanged(app, event)
4956            app.ArrowThickness.FontColor = 'k';
4957
4958        end
4959
4960        % Callback function
4961        function ArrowSizeLegendValueChanged(app, event)
4962            app.ArrowSizeLegend.FontColor = 'k';
4963
4964        end
4965
```

```
4966        % Callback function
4967        function ColorbarTextSize_2ValueChanged(app, event)
4968            app.ColorbarTextSize_2.FontColor = 'k';
4969
4970        end
4971
4972        % Callback function
4973        function ArrowXLabelLocValueChanged(app, event)
4974            app.ArrowXLabelLoc.FontColor = 'k';
4975
4976        end
4977
4978        % Callback function
4979        function ArrowYLabelLocValueChanged(app, event)
4980            app.ArrowYLabelLoc.FontColor = 'k';
4981
4982        end
4983
4984        % Value changed function: MinimumEditField_2
4985        function MinimumEditField_2ValueChanged(app, event)
4986            app.MinimumEditField_2.FontColor = 'k';
4987
4988        end
4989
4990        % Value changed function: MaximumEditField_2
4991        function MaximumEditField_2ValueChanged(app, event)
4992            app.MaximumEditField_2.FontColor = 'k';
4993
4994        end
4995
4996        % Value changed function: WidthEditField_2
4997        function WidthEditField_2ValueChanged(app, event)
4998            app.WidthEditField_2.FontColor = 'k';
4999
5000        end
5001
5002        % Value changed function: SpacingEditField
5003        function SpacingEditFieldValueChanged(app, event)
5004            app.SpacingEditField.FontColor = 'k';
5005
5006        end
5007
5008        % Value changed function: LabelSizeCont
5009        function LabelSizeContValueChanged(app, event)
5010            app.LabelSizeCont.FontColor = 'k';
5011
5012        end
5013
5014        % Value changed function: IntervalEditField_8
5015        function IntervalEditField_8ValueChanged(app, event)
5016            app.IntervalEditField_8.FontColor = 'k';
5017
5018        end
5019
5020        % Value changed function: NorthEditField_2
5021        function NorthEditField_2ValueChanged(app, event)
5022            app.NorthEditField_2.FontColor = 'k';
5023
```

```
5024        end
5025
5026        % Value changed function: WestEditField_2
5027        function WestEditField_2ValueChanged(app, event)
5028            app.WestEditField_2.FontColor = 'k';
5029
5030        end
5031
5032        % Value changed function: EastEditField_2
5033        function EastEditField_2ValueChanged(app, event)
5034            app.EastEditField_2.FontColor = 'k';
5035
5036        end
5037
5038        % Value changed function: SouthEditField_2
5039        function SouthEditField_2ValueChanged(app, event)
5040            app.SouthEditField_2.FontColor = 'k';
5041
5042        end
5043
5044        % Value changed function: Width_3
5045        function Width_3ValueChanged(app, event)
5046            app.Width_3.FontColor = 'k';
5047
5048        end
5049
5050        % Value changed function: Height_3
5051        function Height_3ValueChanged(app, event)
5052            app.Height_3.FontColor = 'k';
5053
5054        end
5055
5056        % Value changed function: OutputDirectoryEditField_3
5057        function OutputDirectoryEditField_3ValueChanged(app, event)
5058            app.OutputDirectoryEditField_3.FontColor = 'k';
5059
5060        end
5061
5062        % Value changed function: FramerateEditField
5063        function FramerateEditFieldValueChanged(app, event)
5064            app.FramerateEditField.FontColor = 'k';
5065
5066        end
5067
5068        % Value changed function: maplabelsize
5069        function maplabelsizeValueChanged(app, event)
5070            app.maplabelsize.FontColor = 'k';
5071
5072        end
5073
5074        % Value changed function: MaxBarValue
5075        function MaxBarValueValueChanged(app, event)
5076            app.MaxBarValue.FontColor = 'k';
5077
5078        end
5079
5080        % Value changed function: MinBarValue
5081        function MinBarValueValueChanged(app, event)
```

```
5082            app.MinBarValue.FontColor = 'k';
5083
5084        end
5085
5086        % Value changed function: InterpolationDivisionEditField_2
5087        function InterpolationDivisionEditField_2ValueChanged(app, event)
5088            app.InterpolationDivisionEditField_2.FontColor = 'k';
5089
5090        end
5091
5092        % Value changed function: FontSizeEditField_2
5093        function FontSizeEditField_2ValueChanged(app, event)
5094            app.FontSizeEditField_2.FontColor = 'k';
5095
5096        end
5097
5098        % Value changed function: SizeEditField_2
5099        function SizeEditField_2ValueChanged(app, event)
5100            app.SizeEditField_2.FontColor = 'k';
5101
5102        end
5103
5104        % Value changed function: FileEditField_2
5105        function FileEditField_2ValueChanged(app, event)
5106
5107        end
5108
5109        % Size changed function: WaveHeightMapTab
5110        function WaveHeightMapTabSizeChanged(app, event)
5111
5112        end
5113
5114        % Button pushed function: CloseFiguresButton_2
5115        function CloseFiguresButton_2Pushed2(app, event)
5116            close all
5117        end
5118
5119        % Value changed function: PlotindegreesCheckBox
5120        function PlotindegreesCheckBoxValueChanged(app, event)
5121
5122        end
5123
5124        % Value changed function: txtCheckBox
5125        function txtCheckBoxValueChanged(app, event)
5126
5127
5128        end
5129
5130        % Value changed function: FlipCheckBox_2
5131        function FlipCheckBox_2ValueChanged(app, event)
5132
5133
5134        end
5135
5136        % Value changed function: YaxisUseDataDropDown
5137        function YaxisUseDataDropDownValueChanged(app, event)
5138            % Determine the Y-axis label based on the dropdown value
5139            switch app.YaxisUseDataDropDown.Value
```

```
5140                 case '1'
5141                     if ~app.CheckBox_4.Value
5142                         app.YAxisEditField.Value = "Time (sec)";
5143                     else %If 'Convert the Values' checkbox is ticked, remove↙
the unit
5144                         app.YAxisEditField.Value = "Time";
5145                     end
5146                 case '2'
5147                     if ~app.CheckBox_4.Value
5148                         app.YAxisEditField.Value = "\eta (m)";
5149                     else
5150                         app.XAxisEditField.Value = "\eta";
5151                     end
5152                 case {'3', '4', '3,4,5'}
5153                     if ~app.CheckBox_4.Value
5154                         app.YAxisEditField.Value = "Velocity (m/s)";
5155                     else
5156                         app.YAxisEditField.Value = "Velocity";
5157                     end
5158             end
5159
5160             %Change the font color of 'Axes Labels: Y axis' in 'Plot Style and↙
Layout'
5161             if ~app.CheckBox_4.Value
5162                 app.YAxisEditField.FontColor = "k";
5163             end
5164
5165             %Update which dropdown list to show in 'Plot Properties: Color' in↙
'Plot Style and Layout'
5166             if length(app.GaugeDirectorylist) == 1 &&  ~strcmp(app.↙
YaxisUseDataDropDown.Value, '3,4,5')
5167                 app.GaugeLineColor.Visible = "on";
5168                 app.GaugeLineColor.Enable = "on";
5169                 app.GaugeLineColor_multiple.Visible = "off";
5170                 app.GaugeLineColor_multiple.Enable = "off";
5171                 app.FlipCheckBox_3.Visible = "off";
5172                 app.FlipCheckBox_3.Enable = "off";
5173             else
5174                 app.GaugeLineColor.Visible = "off";
5175                 app.GaugeLineColor.Enable = "off";
5176                 app.GaugeLineColor_multiple.Visible = "on";
5177                 app.GaugeLineColor_multiple.Enable = "on";
5178                 app.FlipCheckBox_3.Visible = "on";
5179                 app.FlipCheckBox_3.Enable = "on";
5180             end
5181             drawnow
5182         end
5183
5184         % Callback function
5185         function YCol2ValueChanged(app, event)
5186           app.YCol2.FontColor = 'k';
5187
5188         end
5189
5190         % Callback function
5191         function YCol3ValueChanged(app, event)
5192             app.YCol3.FontColor = 'k';
5193
```

```
5194         end
5195
5196         % Callback function
5197         function YCol4ValueChanged(app, event)
5198             app.YCol4.FontColor = 'k';
5199
5200         end
5201
5202         % Size changed function: GaugeRecordsTab
5203         function GaugeRecordsTabSizeChanged(app, event)
5204
5205
5206         end
5207
5208         % Size changed function: Panel_11
5209         function Panel_11SizeChanged(app, event)
5210
5211         end
5212
5213         % Selection changed function: ButtonGroup_16
5214         function ButtonGroup_16SelectionChanged(app, event)
5215             % Enable parameters for the selected basemap option
5216             if ~isempty(app.FileTextArea_3.Value) %Check if there are u_ files↙
loaded in the app
5217                 if app.etaButton.Value
5218                     %Find the corresponding 'eta' file in the directory of the↙
u_ file
5219                     matchingFiles = [];
5220                     for i = 1:length(app.ETAfilenames)
5221                         filename = app.ETAfilenames{i};
5222                         fullPath = fullfile(app.vectorpath, filename);
5223                         if exist(fullPath, 'file') == 2 %Append to the list
5224                             matchingFiles = [matchingFiles; filename];
5225                         end
5226                     end
5227
5228                     if isempty(matchingFiles)
5229                         %No matching files found
5230                         app.FileTextArea_5.Value = 'No matching files found';
5231
5232                         %Disable colormap and colorbar options
5233                         children = app.ThirdTabColorMapPanel.Children;
5234                         for i = 1:length(children)
5235                             children(i).Enable = 'off';
5236                         end
5237
5238                     else  %At least one matching file was found
5239                         app.FileTextArea_5.Value = strjoin(string↙
(matchingFiles), newline);
5240
5241                         %Enable colormap and colorbar options
5242                         children = app.ThirdTabColorMapPanel.Children;
5243                         for i = 1:length(children)
5244                             children(i).Enable = 'on';
5245                         end
5246                         app.BackgroundMapColorDropDown.Value = 'red - blue';
5247                         app.FlipCheckBox_4.Value = true;
5248                     end
```

```
5249
5250                 elseif app.hmaxButton.Value
5251                     %Find the corresponding 'hmax' file in the directory of
the U_ file
5252                     matchingFiles = [];
5253                     for i = 1:length(app.HMAXfilenames)
5254                         filename = app.HMAXfilenames{i};
5255                         fullPath = fullfile(app.vectorpath, filename);
5256                         if exist(fullPath, 'file') == 2
5257                             matchingFiles = [matchingFiles; filename];
5258                         end
5259                     end
5260
5261                     if isempty(matchingFiles)
5262                         % No matching files found
5263                         app.FileTextArea_5.Value = 'No matching files found';
5264
5265                         %Disable colormap and colorbar options
5266                         children = app.ThirdTabColorMapPanel.Children;
5267                         for i = 1:length(children)
5268                             children(i).Enable = 'off';
5269                         end
5270                     else
5271                         % At least one matching file was found
5272                         app.FileTextArea_5.Value = strjoin(string
(matchingFiles), newline);
5273
5274                         %Enable colormap and colorbar options
5275                         children = app.ThirdTabColorMapPanel.Children;
5276                         for i = 1:length(children)
5277                             children(i).Enable = 'on';
5278                         end
5279                         app.BackgroundMapColorDropDown.Value = 'parula';
5280                         app.FlipCheckBox_4.Value = false;
5281                         app.InterpolationDivisionEditField_2.Enable = 'off';
5282                     end
5283
5284                 elseif app.BathymetryButton.Value
5285                     %Use bathymetry data as the basemap
5286                     if isempty(app.DepthFileEditField_2.Value)%If a bathymetry
file is not loaded in the 'Input Data' section
5287                         % No matching files found
5288                         app.FileTextArea_5.Value = 'No bathymetry file found';
drawnow
5289
5290                         %Disable colormap and colorbar options
5291                         children = app.ThirdTabColorMapPanel.Children;
5292                         for i = 1:length(children)
5293                             children(i).Enable = 'off';
5294                         end
5295                         app.LandColor.Enable = 'off';
5296                     else
5297                         %Enable colormap and colorbar options
5298                         children = app.ThirdTabColorMapPanel.Children;
5299                         for i = 1:length(children)
5300                             children(i).Enable = 'on';
5301                         end
5302                          app.FileTextArea_5.Value = ' ';
```

```
5303                            app.BackgroundMapColorDropDown.Value = 'parula';
5304                            app.FlipCheckBox_4.Value = false;
5305                            app.InterpolationDivisionEditField_2.Enable = 'off';
5306                        end
5307
5308                    elseif app.VelocityButton.Value
5309                        children = app.ThirdTabColorMapPanel.Children;
5310                        if ~isempty(app.FileNAME5)
5311                            % Use the Z magnitude, calculated from the U and V↙
vectors, as the basemap
5312                            %Enable colormap and colorbar options
5313                            for i = 1:length(children)
5314                                children(i).Enable = 'on';
5315                            end
5316
5317                            %Change the default colormap and bar limits to match↙
the damage index ranges in Lynett et al. (2022)
5318                            app.BackgroundMapColorDropDown.Value = 'blue -↙
purple';
5319                            app.InterpolationDivisionEditField_2.Value = 4;
5320                            app.InterpolationDivisionEditField_2.Enable = "on";
5321                        else
5322                            for i = 1:length(children)
5323                                children(i).Enable = 'off';
5324                            end
5325                        end
5326
5327                    elseif app.VorticityButton.Value
5328                        children = app.ThirdTabColorMapPanel.Children;
5329                        if ~isempty(app.FileNAME5)
5330                            % Use the Z magnitude, calculated from the U and V↙
vectors, as the basemap
5331                            %Enable colormap and colorbar options
5332                            for i = 1:length(children)
5333                                children(i).Enable = 'on';
5334                            end
5335                            app.BackgroundMapColorDropDown.Value = 'jet';
5336                            app.FlipCheckBox_4.Value = false;
5337                            app.InterpolationDivisionEditField_2.Enable = 'off';
5338                        else
5339                            for i = 1:length(children)
5340                                children(i).Enable = 'off';
5341                            end
5342                        end
5343
5344                    end
5345
5346                    ArrowSection = app.Panel_14.Children;
5347                    if ~app.VorticityButton.Value
5348                        %Enable the Arrow Section
5349                        for i = 1:length(ArrowSection)
5350                            ArrowSection(i).Enable = 'on';
5351                        end
5352                    else
5353                        %Disable the Arrow Section
5354                        for i = 1:length(ArrowSection)
5355                            ArrowSection(i).Enable = 'off';
5356                        end
```

```
5357                    app.PlotvectorsCheckBox.Value = false; %Uncheck the box
5358                end
5359
5360                if contains(string(app.FileNAME5), 'umax')
5361                    for i = 1:length(ArrowSection)
5362                        ArrowSection(i).Enable = 'off';
5363                    end
5364                end
5365            end
5366
5367        end
5368
5369        % Value changed function: LongitudeEditField
5370        function LongitudeEditFieldValueChanged(app, event)
5371            app.LongitudeEditField.FontColor = 'k';
5372
5373            %Update the X-axis limits in the 'General Layout' section
5374            app.x = app.LongitudeEditField.Value + [0:app.col1] * app.gridX.Value;
5375            app.EastEditField.Value = max(app.x);
5376            app.WestEditField.Value = min(app.x);
5377            drawnow
5378        end
5379
5380        % Value changed function: LatitudeEditField
5381        function LatitudeEditFieldValueChanged(app, event)
5382            app.LatitudeEditField.FontColor = 'k';
5383
5384            %Update the Y-axis limits in the 'General Layout' section
5385            app.y = [0:app.row1] * app.gridY.Value+ app.LatitudeEditField.Value;
5386            app.NorthEditField.Value = max(app.y);
5387            app.SouthEditField.Value = min(app.y);
5388            drawnow
5389        end
5390
5391        % Value changed function: gridX
5392        function gridXValueChanged(app, event)
5393            app.gridX.FontColor = 'k';
5394
5395            %Update the X-axis limits in the 'General Layout' section
5396            app.x = app.LongitudeEditField.Value + [0:app.col1] * app.gridX.Value;
5397            app.EastEditField.Value = max(app.x);
5398            app.WestEditField.Value = min(app.x);
5399            drawnow
5400
5401        end
5402
5403        % Value changed function: gridY
5404        function gridYValueChanged(app, event)
5405            app.gridY.FontColor = 'k';
5406
5407            %Update the Y-axis limits in the 'General Layout' section
5408            app.y = [0:app.row1] * app.gridY.Value+ app.LatitudeEditField.Value;
5409            app.NorthEditField.Value = max(app.y);
5410            app.SouthEditField.Value = min(app.y);
```

```
5411            drawnow
5412        end
5413
5414        % Value changed function: TotalSimuilationTimesecEditField_3
5415        function TotalSimuilationTimesecEditField_3ValueChanged(app, event)
5416            app.TotalSimuilationTimesecEditField_3.FontColor = 'k';
5417
5418        end
5419
5420        % Value changed function: tifCheckBox_2
5421        function tifCheckBox_2ValueChanged(app, event)
5422
5423
5424        end
5425
5426        % Callback function
5427        function ArrivalTimeCheckBoxValueChanged(app, event)
5428
5429
5430        end
5431
5432        % Callback function
5433        function TimeFileButtonPushed(app, event)
5434            [timefile,path2] = uigetfile('*.*', 'Select the files',↙
'MultiSelect', 'on'); %uigetfile('*.txt;*.out'); %;*.tiff;*.tif;*.asdf;*.asc;*.grd;↙
*.flt;*.grb;*.grid;*.grib2;*.dt0;*.dt1;*.dt2;*.ddf;*.dem;*.hgt;*.grd;*.grc;*.ers;*.↙
dat;*.img;*.jp2'); uigetfile('*.txt;*.shp')
5435            app.FileEditField_3.Value = string(timefile);
5436            app.FileEditField_3.FontColor = 'k';
5437
5438            if ~(ismcc || isdeployed)
5439                addpath(genpath(fullfile(string(path2))));
5440            end
5441
5442            conditions = [
5443                ~strcmp(app.FileEditField_3.Value, "0"),
5444                ~strcmp(app.FileEditField_3.Value, "00")
5445                ];
5446
5447            if app.FileEditField_3.Value == string(timefile)
5448                if any(conditions)
5449                    try
5450                        ArrivalTimeinputdata = string(path2) + string↙
(timefile);
5451                        app.ArrivalTimename = string(timefile);
5452                        app.FileEditField_3.BackgroundColor = 'w';
5453                        app.FileEditField_3.FontColor =  'k';
5454                        ArrivalTimeDATA = readmatrix(ArrivalTimeinputdata);
5455                        app.ArrivalTimeConverted =  ArrivalTimeDATA ./ 60; %↙
convert the values to minutes. Default threshold to read first wave arrival time in↙
FUNWAVE is 0.001 m
5456                        app.MaximumdepthEditField_3.Value = max(max(app.↙
ArrivalTimeConverted));
5457                        app.MaximumdepthEditField_3.FontColor = 'k';
5458                    end
5459                end
5460            end
5461
```

```
5462          end
5463
5464          % Callback function
5465          function AddLabelsCheckBox_5ValueChanged(app, event)
5466              if app.AddLabelsCheckBox_5.Value
5467                  app.SpacingEditField_4Label_6.Enable = "on";
5468                  app.TextLabelSize_3.Enable = "on";
5469                  app.LabelSpacingEditField_2Label_3.Enable = "on";
5470                  app.LabelSpacingEditField_4.Enable = "on";
5471                  app.LabelSpacingEditField_2Label_4.Enable = "on";
5472                  app.IntervalEditField_7.Enable = "on";
5473              else
5474                  app.SpacingEditField_4Label_6.Enable = "off";
5475                  app.TextLabelSize_3.Enable = "off";
5476                  app.LabelSpacingEditField_2Label_3.Enable = "off";
5477                  app.LabelSpacingEditField_4.Enable = "off";
5478                  app.LabelSpacingEditField_2Label_4.Enable = "off";
5479                  app.IntervalEditField_7.Enable = "off";
5480              end
5481
5482          end
5483
5484          % Value changed function: XaxisUseDataDropDown
5485          function XaxisUseDataDropDownValueChanged(app, event)
5486              % Determine the X-axis label based on the dropdown value
5487              switch app.XaxisUseDataDropDown.Value
5488                  case '1'
5489                      if ~app.CheckBox_3.Value
5490                          app.XAxisEditField.Value = "Time (sec)";
5491                      else %If 'Convert the Values' checkbox is ticked, remove↙
the unit
5492                          app.XAxisEditField.Value = "Time";
5493                      end
5494                  case '2'
5495                      if ~app.CheckBox_3.Value
5496                          app.XAxisEditField.Value = "\eta (m)";
5497                      else
5498                          app.XAxisEditField.Value = "\eta";
5499                      end
5500                  case {'3', '4'}  % Handles both '3' and '4'
5501                      if ~app.CheckBox_3.Value
5502                          app.XAxisEditField.Value = "Velocity (m/s)";
5503                      else
5504                          app.XAxisEditField.Value = "Velocity";
5505                      end
5506              end
5507
5508              %Change the font color of 'Axes Labels: X axis' in 'Plot Style and↙
Layout'
5509              if ~app.CheckBox_3.Value
5510                  app.XAxisEditField.FontColor = "k";
5511              end
5512
5513
5514          end
5515
5516          % Selection changed function: ButtonGroup_17
5517          function ButtonGroup_17SelectionChanged(app, event)
```

```
5518            %Display parameter options for each tab when selected
5519            if app.BathymetryButton_2.Value
5520                app.TabGroup2.SelectedTab = app.BathymetryTab;
5521            elseif app.WaveHeightButton.Value
5522                app.TabGroup2.SelectedTab = app.WaveHeightTab;
5523            elseif app.ArrivalTimeButton.Value
5524                app.TabGroup2.SelectedTab = app.ArrivalTimeTab;
5525            end
5526
5527            tabs = {app.BathymetryTab, app.WaveHeightTab, app.ArrivalTimeTab};
5528            components1 = {app.LabelSpacingEditField, app.IntervalEditField_3, ↙
app.TextLabelSize,   app.LabelSpacingEditFieldLabel_2, app.↙
LabelSpacingEditFieldLabel, app.SpacingEditField_4Label_4};
5529            components2 = {app.LabelSpacingEditField_2, app.wh_interval, app.↙
TextLabelSize_2, app.LabelSpacingEditField_2Label_2, app.↙
LabelSpacingEditField_2Label, app.SpacingEditField_4Label_5};
5530            components3 = {app.LabelSpacingEditField_2Label_3, app.↙
LabelSpacingEditField_4, app.SpacingEditField_4Label_6, app.TextLabelSize_3, app.↙
IntervalEditField_7, app.LabelSpacingEditField_2Label_4};
5531
5532            for k = 1:length(tabs)
5533                children = tabs{k}.Children;
5534                for i = 1:length(children)
5535                    if app.NoneButton.Value   %Disable all the items when↙
'None' is selected
5536                        children(i).Enable = 'off';
5537                    else  %Enable all the items
5538                        children(i).Enable = 'on';
5539
5540                        %For 'Add Labels' suboptions, enable only when the box↙
is ticked
5541                        %Bathymetry Tab
5542                        for i = 1:length(components1)
5543                            if  app.AddLabelsCheckBox_2.Value
5544                                components1{i}.Enable = 'on';
5545                            else
5546                                components1{i}.Enable = 'off';
5547                            end
5548                        end
5549
5550                        %Wave Height Tab
5551                        for i = 1:length(components2)
5552                            if app.AddLabelsCheckBox_3.Value
5553                                components2{i}.Enable = 'on';
5554                            else
5555                                components2{i}.Enable = 'off';
5556                            end
5557                        end
5558
5559                        %Arrival Time Tab
5560                        for i = 1:length(components3)
5561                            if app.AddLabelsCheckBox_5.Value
5562                                components3{i}.Enable = 'on';
5563                            else
5564                                components3{i}.Enable = 'off';
5565                            end
5566                        end
5567
```

```
5568                    end
5569                end
5570            end
5571        end
5572
5573        % Size changed function: ButtonGroup_17
5574        function ButtonGroup_17SizeChanged(app, event)
5575
5576        end
5577
5578        % Callback function
5579        function LineintervalEditField_3ValueChanged(app, event)
5580            app.IntervalEditField_7.Value =   app.LineintervalEditField_3.↙
Value;
5581            app.LineintervalEditField_3.FontColor = 'k';
5582
5583        end
5584
5585        % Callback function
5586        function LabelSpacingEditField_4ValueChanged(app, event)
5587            app.LabelSpacingEditField_4.FontColor = 'k';
5588
5589        end
5590
5591        % Callback function
5592        function TextLabelSize_3ValueChanged(app, event)
5593            app.TextLabelSize_3.FontColor = 'k';
5594
5595        end
5596
5597        % Callback function
5598        function ThicknessEditField_3ValueChanged(app, event)
5599            app.ThicknessEditField_3.FontColor = 'k';
5600
5601        end
5602
5603        % Callback function
5604        function MinimumdepthEditField_3ValueChanged(app, event)
5605            app.MinimumdepthEditField_3.FontColor = 'k';
5606
5607        end
5608
5609        % Callback function
5610        function MaximumdepthEditField_3ValueChanged(app, event)
5611            app.MaximumdepthEditField_3.FontColor = 'k';
5612
5613        end
5614
5615        % Drop down opening function: YaxisUseDataDropDown
5616        function YaxisUseDataDropDownOpening(app, event)
5617
5618        end
5619
5620        % Value changed function: ArrowSpacing
5621        function ArrowSpacingValueChanged(app, event)
5622            app.ArrowSpacing.FontColor = 'k';
5623
5624        end
```

```
5625
5626        % Value changed function: StartTime
5627        function StartTimeValueChanged(app, event)
5628            app.StartTime.FontColor = 'k';
5629        end
5630
5631        % Value changed function: ColorbarTextSize
5632        function ColorbarTextSizeValueChanged(app, event)
5633            app.ColorbarTextSize.FontColor = 'k';
5634
5635        end
5636
5637        % Value changed function: FlipCheckBox_5
5638        function FlipCheckBox_5ValueChanged(app, event)
5639
5640        end
5641
5642        % Value changed function: StartTime2
5643        function StartTime2ValueChanged(app, event)
5644
5645
5646        end
5647
5648        % Value changed function: PlotvectorsCheckBox
5649        function PlotvectorsCheckBoxValueChanged(app, event)
5650
5651
5652        end
5653
5654        % Value changed function: GLspacing_2
5655        function GLspacing_2ValueChanged(app, event)
5656       app.GLspacing_2.FontColor = 'k';
5657
5658        end
5659
5660        % Value changing function: XAxisEditField
5661        function XAxisEditFieldValueChanging(app, event)
5662
5663        end
5664
5665        % Value changed function: legendfirsttext
5666        function legendfirsttextValueChanged(app, event)
5667         app.legendfirsttext.FontColor = 'k';
5668
5669        end
5670
5671        % Value changed function: LineYThickness_2
5672        function LineYThickness_2ValueChanged(app, event)
5673         app.LineYThickness_2.FontColor = 'k';
5674
5675        end
5676
5677        % Value changed function: Width
5678        function WidthValueChanged(app, event)
5679           app.Width.FontColor = 'k';
5680
5681        end
5682
```

```
5683        % Value changed function: Height
5684        function HeightValueChanged(app, event)
5685            app.Height.FontColor = 'k';
5686
5687        end
5688
5689        % Value changed function: Width_2
5690        function Width_2ValueChanged(app, event)
5691            app.Width_2.FontColor = 'k';
5692
5693        end
5694
5695        % Value changed function: Height_2
5696        function Height_2ValueChanged(app, event)
5697            app.Height_2.FontColor = 'k';
5698
5699        end
5700
5701        % Value changed function: OutputDirectoryEditField_2
5702        function OutputDirectoryEditField_2ValueChanged(app, event)
5703
5704
5705        end
5706
5707        % Value changed function: mp4CheckBox_2
5708        function mp4CheckBox_2ValueChanged(app, event)
5709            if app.mp4CheckBox_2.Value
5710                app.FramerateEditFieldLabel.Visible = "on";
5711                app.FramerateEditField.Visible = "on";
5712            else
5713                app.FramerateEditFieldLabel.Visible = "off";
5714                app.FramerateEditField.Visible = "off";
5715            end
5716
5717        end
5718
5719        % Callback function
5720        function IntervalEditField_7ValueChanged(app, event)
5721            app.IntervalEditField_7.FontColor = 'k';
5722        end
5723
5724        % Button pushed function: TimeFileButton
5725        function TimeFileButtonPushed2(app, event)
5726            [timefile,path2] = uigetfile('*.*', 'Select the files',↙
'MultiSelect', 'on'); %uigetfile('*.txt;*.out'); %;*.tiff;*.tif;*.asdf;*.asc;*.grd;↙
*.flt;*.grb;*.grid;*.grib2;*.dt0;*.dt1;*.dt2;*.ddf;*.dem;*.hgt;*.grd;*.grc;*.ers;*.↙
dat;*.img;*.jp2'); uigetfile('*.txt;*.shp')
5727            app.FileEditField_3.Value = string(timefile);
5728            app.FileEditField_3.FontColor = 'k';
5729
5730            if ~(ismcc || isdeployed)
5731                addpath(genpath(fullfile(string(path2))));
5732            end
5733
5734            conditions = [
5735                ~strcmp(app.FileEditField_3.Value, "0"),
5736                ~strcmp(app.FileEditField_3.Value, "00")
5737                ];
```

```
5738
5739             if app.FileEditField_3.Value == string(timefile)
5740                 if any(conditions)
5741                     try
5742                         ArrivalTimeinputdata = string(path2) + string↙
(timefile);
5743                         app.ArrivalTimename = string(timefile);
5744                         app.FileEditField_3.BackgroundColor = 'w';
5745                         app.FileEditField_3.FontColor =  'k';
5746                         ArrivalTimeDATA = readmatrix(ArrivalTimeinputdata);
5747                         app.ArrivalTimeConverted =  round(ArrivalTimeDATA ./↙
60, 2); %convert the values to minutes. Default threshold to read first wave↙
arrival time in FUNWAVE is 0.001 m
5748                         app.MaximumdepthEditField_3.Value = max(max(app.↙
ArrivalTimeConverted));
5749                         app.MaximumdepthEditField_3.FontColor = 'k';
5750                     end
5751                 end
5752             end
5753
5754         end
5755
5756         % Button pushed function: Button_8
5757         function Button_8Pushed2(app, event)
5758
5759             [filename3,path3] = uigetfile('*.txt;*.shp');
5760             fullname = fullfile(path3,filename3);
5761
5762             [~, name, ext] = fileparts(filename3);
5763             if ~(ismcc || isdeployed)
5764                 addpath(genpath(string(path3)));
5765             end
5766
5767             app.GaugeFname = name;
5768             app.GAUGEFILE = filename3;
5769             app.FileEditField.Value = filename3;
5770             app.FileEditField.FontColor = 'k';
5771             app.FileEditField.BackgroundColor = 'w';
5772
5773
5774             if strcmp(ext, '.shp')
5775                 S = shaperead(fullname);
5776                 latitudes= [S.Y]';
5777                 longitudes= [S.X]';
5778                 list = [round(latitudes, 4), round(longitudes, 4)];
5779                 list = unique(list, 'rows');
5780                 app.latGauge = list(:, 1);
5781                 app.longGauge =  list(:, 2);
5782             elseif strcmp(ext, '.txt')
5783                 fileID = fopen(fullname, 'r');
5784                 dataArray = textscan(fileID, '%f %f', 'Delimiter',↙
'whitespace'); %Read text file and ensure that it is tab delimited
5785                 fclose(fileID);
5786                 app.STATION_FILE = string(filename3);
5787                 app.latGauge  = dataArray{1};
5788                 app.longGauge = dataArray{2};
5789             end
5790
```

```
5791        end
5792
5793        % Value changed function: AddLabelsCheckBox_4
5794        function AddLabelsCheckBox_4ValueChanged2(app, event)
5795
5796               if app.AddLabelsCheckBox_4.Value
5797                app.CoastlinecolourDropDown_5.Enable = "on";
5798                app.CoastlinecolourDropDown_4.Enable = "on";
5799                app.SizeEditField_3Label.Enable = "on";
5800                app.AlignmentLabel.Enable = "on";
5801                app.SizeEditField_3.Enable = "on";
5802                app.SizeEditField_3.Editable = "on";
5803                app.GLspacing.Enable = "on";
5804                app.GLspacing.Editable = "on";
5805                app.GLspacingLabel.Enable = "on";
5806                app.FontSizeEditField_2Label.Enable = "on";
5807                app.AlignmentLabel_2.Enable = "on";
5808            else
5809                app.CoastlinecolourDropDown_5.Enable = "off";
5810                app.CoastlinecolourDropDown_4.Enable = "off";
5811                app.SizeEditField_3Label.Enable = "off";
5812                app.AlignmentLabel.Enable = "off";
5813                app.SizeEditField_3.Enable = "off";
5814                app.SizeEditField_3.Editable = "off";
5815                app.GLspacing.Enable = "off";
5816                app.GLspacing.Editable = "off";
5817                app.GLspacingLabel.Enable = "off";
5818                app.FontSizeEditField_2Label.Enable = "off";
5819                app.AlignmentLabel_2.Enable = "off";
5820            end
5821        end
5822
5823        % Value changed function: AddLabelsCheckBox_5
5824        function AddLabelsCheckBox_5ValueChanged2(app, event)
5825            if app.AddLabelsCheckBox_5.Value
5826                app.SpacingEditField_4Label_6.Enable = "on";
5827                app.TextLabelSize_3.Enable = "on";
5828                app.LabelSpacingEditField_2Label_3.Enable = "on";
5829                app.LabelSpacingEditField_4.Enable = "on";
5830                app.LabelSpacingEditField_2Label_4.Enable = "on";
5831                app.IntervalEditField_7.Enable = "on";
5832            else
5833                app.SpacingEditField_4Label_6.Enable = "off";
5834                app.TextLabelSize_3.Enable = "off";
5835                app.LabelSpacingEditField_2Label_3.Enable = "off";
5836                app.LabelSpacingEditField_4.Enable = "off";
5837                app.LabelSpacingEditField_2Label_4.Enable = "off";
5838                app.IntervalEditField_7.Enable = "off";
5839            end
5840        end
5841
5842        % Value changed function: AddLabelsCheckBox_3
5843        function AddLabelsCheckBox_3ValueChanged2(app, event)
5844            if app.AddLabelsCheckBox_3.Value
5845                app.TextLabelSize_2.Enable = "on";
5846                app.wh_interval.Enable = "on";
5847                app.LabelSpacingEditField_2.Enable = "on";
5848                app.SpacingEditField_4Label_5.Enable = "on";
```

```
5849                 app.LabelSpacingEditField_2Label_2.Enable = "on";
5850                 app.LabelSpacingEditField_2Label.Enable = "on";
5851             else
5852                 app.TextLabelSize_2.Enable = "off";
5853                 app.wh_interval.Enable = "off";
5854                 app.LabelSpacingEditField_2.Enable = "off";
5855                 app.SpacingEditField_4Label_5.Enable =  "off";
5856                 app.LabelSpacingEditField_2Label_2.Enable =  "off";
5857                 app.LabelSpacingEditField_2Label.Enable =  "off";
5858             end
5859         end
5860 
5861         % Value changed function: AddLabelsCheckBox_2
5862         function AddLabelsCheckBox_2ValueChanged2(app, event)
5863              if app.AddLabelsCheckBox_2.Value
5864                 app.IntervalEditField_3.Enable = "on";
5865                 app.TextLabelSize.Enable = "on";
5866                 app.LabelSpacingEditField.Enable = "on";
5867                 app.SpacingEditField_4Label_4.Enable = "on";
5868                 app.LabelSpacingEditFieldLabel.Enable = "on";
5869                 app.LabelSpacingEditFieldLabel_2.Enable = "on";
5870             else
5871                 app.IntervalEditField_3.Enable = "off";
5872                 app.TextLabelSize.Enable = "off";
5873                 app.LabelSpacingEditField.Enable = "off";
5874                 app.SpacingEditField_4Label_4.Enable = "off";
5875                 app.LabelSpacingEditFieldLabel.Enable = "off";
5876                 app.LabelSpacingEditFieldLabel_2.Enable = "off";
5877             end
5878         end
5879     end
5880 
5881     % Component initialization
5882     methods (Access = private)
5883 
5884         % Create UIFigure and components
5885         function createComponents(app)
5886 
5887             % Create UIFigure and hide until all components are created
5888             app.UIFigure = uifigure('Visible', 'off');
5889             app.UIFigure.AutoResizeChildren = 'off';
5890             app.UIFigure.Position = [1 11 542 861];
5891             app.UIFigure.Name = 'MATLAB App';
5892             app.UIFigure.Scrollable = 'on';
5893 
5894             % Create TabGroup
5895             app.TabGroup = uitabgroup(app.UIFigure);
5896             app.TabGroup.AutoResizeChildren = 'off';
5897             app.TabGroup.Position = [1 1 542 861];
5898 
5899             % Create WaveHeightMapTab
5900             app.WaveHeightMapTab = uitab(app.TabGroup);
5901             app.WaveHeightMapTab.AutoResizeChildren = 'off';
5902             app.WaveHeightMapTab.SizeChangedFcn = createCallbackFcn(app, ↙
@WaveHeightMapTabSizeChanged, true);
5903             app.WaveHeightMapTab.Title = 'Wave Height Map';
5904             app.WaveHeightMapTab.Scrollable = 'on';
5905 
```

```
5906             % Create Panel_3
5907             app.Panel_3 = uipanel(app.WaveHeightMapTab);
5908             app.Panel_3.AutoResizeChildren = 'off';
5909             app.Panel_3.ForegroundColor = [0.149 0.149 0.149];
5910             app.Panel_3.Position = [11 662 520 160];
5911
5912             % Create FileTextArea
5913             app.FileTextArea = uitextarea(app.Panel_3);
5914             app.FileTextArea.ValueChangedFcn = createCallbackFcn(app, ↙
@FileTextAreaValueChanged, true);
5915             app.FileTextArea.Editable = 'off';
5916             app.FileTextArea.FontColor = [0.149 0.149 0.149];
5917             app.FileTextArea.Tooltip = {''};
5918             app.FileTextArea.Placeholder = 'eta_xxxx; hmax_xxxx;';
5919             app.FileTextArea.Position = [43 65 95 55];
5920
5921             % Create Button_14
5922             app.Button_14 = uibutton(app.Panel_3, 'push');
5923             app.Button_14.ButtonPushedFcn = createCallbackFcn(app, ↙
@Button_14Pushed, true);
5924             app.Button_14.FontAngle = 'italic';
5925             app.Button_14.FontColor = [0.651 0.651 0.651];
5926             app.Button_14.Tooltip = {'Load the files'; ''; 'Files must start↙
with ''eta'' or ''hmax'''};
5927             app.Button_14.Position = [146 101 19 19];
5928             app.Button_14.Text = '...';
5929
5930             % Create SouthwestCornerLabel
5931             app.SouthwestCornerLabel = uilabel(app.Panel_3);
5932             app.SouthwestCornerLabel.FontWeight = 'bold';
5933             app.SouthwestCornerLabel.Tooltip = {'Coordinates of the southwest↙
corner of the loaded files.'};
5934             app.SouthwestCornerLabel.Position = [232 121 110 22];
5935             app.SouthwestCornerLabel.Text = 'Southwest Corner';
5936
5937             % Create LongitudeEditFieldLabel
5938             app.LongitudeEditFieldLabel = uilabel(app.Panel_3);
5939             app.LongitudeEditFieldLabel.HorizontalAlignment = 'right';
5940             app.LongitudeEditFieldLabel.WordWrap = 'on';
5941             app.LongitudeEditFieldLabel.Tooltip = {'West boundary'; ''; 'Unit:↙
degrees / meters'};
5942             app.LongitudeEditFieldLabel.Position = [228 98 60 22];
5943             app.LongitudeEditFieldLabel.Text = 'Longitude';
5944
5945             % Create LongitudeEditField
5946             app.LongitudeEditField = uieditfield(app.Panel_3, 'numeric');
5947             app.LongitudeEditField.ValueDisplayFormat = '%8.4f';
5948             app.LongitudeEditField.ValueChangedFcn = createCallbackFcn(app, ↙
@LongitudeEditFieldValueChanged, true);
5949             app.LongitudeEditField.FontColor = [0.651 0.651 0.651];
5950             app.LongitudeEditField.Tooltip = {''};
5951             app.LongitudeEditField.Position = [296 104 60 16];
5952
5953             % Create LatitudeEditFieldLabel
5954             app.LatitudeEditFieldLabel = uilabel(app.Panel_3);
5955             app.LatitudeEditFieldLabel.HorizontalAlignment = 'right';
5956             app.LatitudeEditFieldLabel.WordWrap = 'on';
5957             app.LatitudeEditFieldLabel.Tooltip = {'South boundary'; ''; 'Unit:↙
```

```
degrees / meters'};
5958             app.LatitudeEditFieldLabel.Position = [234 81 44 17];
5959             app.LatitudeEditFieldLabel.Text = 'Latitude';
5960
5961             % Create LatitudeEditField
5962             app.LatitudeEditField = uieditfield(app.Panel_3, 'numeric');
5963             app.LatitudeEditField.ValueDisplayFormat = '%8.4f';
5964             app.LatitudeEditField.ValueChangedFcn = createCallbackFcn(app, ↙
@LatitudeEditFieldValueChanged, true);
5965             app.LatitudeEditField.FontColor = [0.651 0.651 0.651];
5966             app.LatitudeEditField.Tooltip = {''};
5967             app.LatitudeEditField.Position = [296 82 60 16];
5968
5969             % Create Button_15
5970             app.Button_15 = uibutton(app.Panel_3, 'push');
5971             app.Button_15.ButtonPushedFcn = createCallbackFcn(app, ↙
@Button_15Pushed, true);
5972             app.Button_15.FontAngle = 'italic';
5973             app.Button_15.FontColor = [0.651 0.651 0.651];
5974             app.Button_15.Tooltip = {'Load the file'; ''; 'Select the↙
bathymetry used in the simulation, or the output ''mask_xxxx'' file.'; ''; 'Format↙
accepted: .txt, .tif, mask_'};
5975             app.Button_15.Position = [146 11 19 19];
5976             app.Button_15.Text = '...';
5977
5978             % Create DepthFileEditField
5979             app.DepthFileEditField = uieditfield(app.Panel_3, 'text');
5980             app.DepthFileEditField.ValueChangedFcn = createCallbackFcn(app, ↙
@DepthFileEditFieldValueChanged, true);
5981             app.DepthFileEditField.Editable = 'off';
5982             app.DepthFileEditField.FontColor = [0.149 0.149 0.149];
5983             app.DepthFileEditField.Tooltip = {''};
5984             app.DepthFileEditField.Placeholder = '.txt, .tif., mask_';
5985             app.DepthFileEditField.Position = [44 13 96 16];
5986
5987             % Create FilesLabel
5988             app.FilesLabel = uilabel(app.Panel_3);
5989             app.FilesLabel.FontWeight = 'bold';
5990             app.FilesLabel.Tooltip = {'Load the files'; ''; 'Files must start↙
with ''eta'' or ''hmax'''};
5991             app.FilesLabel.Position = [44 121 32 22];
5992             app.FilesLabel.Text = 'Files';
5993
5994             % Create BathymetryLabel
5995             app.BathymetryLabel = uilabel(app.Panel_3);
5996             app.BathymetryLabel.FontWeight = 'bold';
5997             app.BathymetryLabel.Tooltip = {'Load the file'; ''; 'Select the↙
bathymetry used in the simulation, or the output ''mask_xxxx'' file.'; ''; 'Format↙
accepted: .txt, .tif, mask_'};
5998             app.BathymetryLabel.Position = [44 31 71 22];
5999             app.BathymetryLabel.Text = 'Bathymetry';
6000
6001             % Create TotalSimuilationTimesecEditFieldLabel_3
6002             app.TotalSimuilationTimesecEditFieldLabel_3 = uilabel(app.↙
Panel_3);
6003             app.TotalSimuilationTimesecEditFieldLabel_3.FontWeight = 'bold';
6004             app.TotalSimuilationTimesecEditFieldLabel_3.Tooltip = {'For values↙
>60 seconds, the unit will be automatically converted to minutes or hours. The user↙
```

```
can verify whether the conversion is correct by double-checking the LOG.TXT file generated after the simulation.'; ''; 'For example, in the line:'; '"PRINTING FILE NO. 1  TIME/TOTAL: 3600.000 / 7200.000"'; 'the hmax or eta 00001 entry will be labelled with a title corresponding to 1 hour.'};
            app.TotalSimuilationTimesecEditFieldLabel_3.Position = [232 31 98 22];
            app.TotalSimuilationTimesecEditFieldLabel_3.Text = 'Simulation Time';

            % Create TotalSimuilationTimesecEditField_3
            app.TotalSimuilationTimesecEditField_3 = uieditfield(app.Panel_3, 'numeric');
            app.TotalSimuilationTimesecEditField_3.Limits = [1 Inf];
            app.TotalSimuilationTimesecEditField_3.ValueDisplayFormat = '%8.1f';
            app.TotalSimuilationTimesecEditField_3.ValueChangedFcn = createCallbackFcn(app, @TotalSimuilationTimesecEditField_3ValueChanged, true);
            app.TotalSimuilationTimesecEditField_3.FontColor = [0.651 0.651 0.651];
            app.TotalSimuilationTimesecEditField_3.Tooltip = {''};
            app.TotalSimuilationTimesecEditField_3.Position = [409 16 60 16];
            app.TotalSimuilationTimesecEditField_3.Value = 1;

            % Create IntervalLabel
            app.IntervalLabel = uilabel(app.Panel_3);
            app.IntervalLabel.HorizontalAlignment = 'right';
            app.IntervalLabel.Tooltip = {'Snapshot interval.'; ''; 'Same as PLOT_INTV in input.txt'; ''; 'Unit: seconds'};
            app.IntervalLabel.Position = [358 13 45 22];
            app.IntervalLabel.Text = 'Interval';

            % Create StartTime
            app.StartTime = uieditfield(app.Panel_3, 'numeric');
            app.StartTime.Limits = [0 Inf];
            app.StartTime.ValueDisplayFormat = '%8.1f';
            app.StartTime.ValueChangedFcn = createCallbackFcn(app, @StartTimeValueChanged, true);
            app.StartTime.FontColor = [0.651 0.651 0.651];
            app.StartTime.Tooltip = {''};
            app.StartTime.Position = [296 16 47 16];

            % Create StartLabel
            app.StartLabel = uilabel(app.Panel_3);
            app.StartLabel.HorizontalAlignment = 'right';
            app.StartLabel.Tooltip = {'For imported initial tsunami files with a non-zero start time, input the value here. Example: for a 30-second snapshot, input ''30''.'; ''; 'Unit: seconds'};
            app.StartLabel.Position = [255 13 31 22];
            app.StartLabel.Text = 'Start';

            % Create GridSizeLabel_2
            app.GridSizeLabel_2 = uilabel(app.Panel_3);
            app.GridSizeLabel_2.FontWeight = 'bold';
            app.GridSizeLabel_2.Tooltip = {'Resolution'; 'Unit: degrees / meters'};
            app.GridSizeLabel_2.Position = [386 121 57 22];
            app.GridSizeLabel_2.Text = 'Grid Size';

```

```
6048            % Create XEditField_2Label
6049            app.XEditField_2Label = uilabel(app.Panel_3);
6050            app.XEditField_2Label.HorizontalAlignment = 'center';
6051            app.XEditField_2Label.Tooltip = {'X direction'; ''; 'Unit: degrees↙
/ meters'};
6052            app.XEditField_2Label.Position = [388 98 13 22];
6053            app.XEditField_2Label.Text = 'X ';
6054
6055            % Create gridX
6056            app.gridX = uieditfield(app.Panel_3, 'numeric');
6057            app.gridX.Limits = [0 Inf];
6058            app.gridX.ValueDisplayFormat = '%8.5f';
6059            app.gridX.ValueChangedFcn = createCallbackFcn(app,↙
@gridXValueChanged, true);
6060            app.gridX.FontColor = [0.651 0.651 0.651];
6061            app.gridX.Tooltip = {''};
6062            app.gridX.Position = [409 104 60 16];
6063
6064            % Create YEditField_2Label
6065            app.YEditField_2Label = uilabel(app.Panel_3);
6066            app.YEditField_2Label.HorizontalAlignment = 'center';
6067            app.YEditField_2Label.Tooltip = {'Y direction'; ''; 'Unit: degrees↙
/ meters'};
6068            app.YEditField_2Label.Position = [386 79 13 21];
6069            app.YEditField_2Label.Text = 'Y';
6070
6071            % Create gridY
6072            app.gridY = uieditfield(app.Panel_3, 'numeric');
6073            app.gridY.Limits = [0 Inf];
6074            app.gridY.ValueDisplayFormat = '%8.5f';
6075            app.gridY.ValueChangedFcn = createCallbackFcn(app,↙
@gridYValueChanged, true);
6076            app.gridY.FontColor = [0.651 0.651 0.651];
6077            app.gridY.Tooltip = {''};
6078            app.gridY.Position = [409 82 60 16];
6079
6080            % Create InputDataLabel
6081            app.InputDataLabel = uilabel(app.WaveHeightMapTab);
6082            app.InputDataLabel.BackgroundColor = [0.9412 0.9412 0.9412];
6083            app.InputDataLabel.FontSize = 15;
6084            app.InputDataLabel.FontWeight = 'bold';
6085            app.InputDataLabel.FontColor = [0.0314 0.3686 0.6];
6086            app.InputDataLabel.Tooltip = {''};
6087            app.InputDataLabel.Position = [38 813 97 22];
6088            app.InputDataLabel.Text = '  Input Data  ';
6089
6090            % Create Panel_7
6091            app.Panel_7 = uipanel(app.WaveHeightMapTab);
6092            app.Panel_7.AutoResizeChildren = 'off';
6093            app.Panel_7.TitlePosition = 'righttop';
6094            app.Panel_7.FontSize = 10;
6095            app.Panel_7.Position = [11 315 520 168];
6096
6097            % Create TabGroup2
6098            app.TabGroup2 = uitabgroup(app.Panel_7);
6099            app.TabGroup2.AutoResizeChildren = 'off';
6100            app.TabGroup2.Position = [0 0 520 153];
6101
```

```
6102             % Create BathymetryTab
6103             app.BathymetryTab = uitab(app.TabGroup2);
6104             app.BathymetryTab.AutoResizeChildren = 'off';
6105             app.BathymetryTab.Title = 'Bathymetry';
6106 
6107             % Create MaximumEditFieldLabel
6108             app.MaximumEditFieldLabel = uilabel(app.BathymetryTab);
6109             app.MaximumEditFieldLabel.Enable = 'off';
6110             app.MaximumEditFieldLabel.Tooltip = {'Higher value indicate deeper↙
waters'};
6111             app.MaximumEditFieldLabel.Position = [48 25 58 22];
6112             app.MaximumEditFieldLabel.Text = 'Maximum';
6113 
6114             % Create MaximumEditField
6115             app.MaximumEditField = uieditfield(app.BathymetryTab, 'numeric');
6116             app.MaximumEditField.Limits = [1 Inf];
6117             app.MaximumEditField.ValueDisplayFormat = '%8.1f';
6118             app.MaximumEditField.Enable = 'off';
6119             app.MaximumEditField.Tooltip = {''};
6120             app.MaximumEditField.Position = [107 28 50 16];
6121             app.MaximumEditField.Value = 1000;
6122 
6123             % Create MinimumEditFieldLabel
6124             app.MinimumEditFieldLabel = uilabel(app.BathymetryTab);
6125             app.MinimumEditFieldLabel.Enable = 'off';
6126             app.MinimumEditFieldLabel.Tooltip = {'0 value = coastline'; '';↙
'Lower values indicate shallower waters.'};
6127             app.MinimumEditFieldLabel.Position = [48 71 55 22];
6128             app.MinimumEditFieldLabel.Text = 'Minimum';
6129 
6130             % Create MinimumEditField
6131             app.MinimumEditField = uieditfield(app.BathymetryTab, 'numeric');
6132             app.MinimumEditField.Limits = [0 Inf];
6133             app.MinimumEditField.ValueDisplayFormat = '%8.1f';
6134             app.MinimumEditField.Enable = 'off';
6135             app.MinimumEditField.Tooltip = {''};
6136             app.MinimumEditField.Position = [107 74 50 16];
6137 
6138             % Create IntervalEditField_5Label
6139             app.IntervalEditField_5Label = uilabel(app.BathymetryTab);
6140             app.IntervalEditField_5Label.Enable = 'off';
6141             app.IntervalEditField_5Label.Tooltip = {'Interval between adjacent↙
contours'};
6142             app.IntervalEditField_5Label.Position = [48 48 45 22];
6143             app.IntervalEditField_5Label.Text = 'Interval';
6144 
6145             % Create IntervalEditField_5
6146             app.IntervalEditField_5 = uieditfield(app.BathymetryTab,↙
'numeric');
6147             app.IntervalEditField_5.Limits = [0 Inf];
6148             app.IntervalEditField_5.ValueDisplayFormat = '%8.1f';
6149             app.IntervalEditField_5.Enable = 'off';
6150             app.IntervalEditField_5.Tooltip = {''};
6151             app.IntervalEditField_5.Position = [107 51 40 17];
6152             app.IntervalEditField_5.Value = 500;
6153 
6154             % Create DepthRangeLabel
6155             app.DepthRangeLabel = uilabel(app.BathymetryTab);
```

```
6156            app.DepthRangeLabel.FontWeight = 'bold';
6157            app.DepthRangeLabel.Enable = 'off';
6158            app.DepthRangeLabel.Position = [48 95 80 22];
6159            app.DepthRangeLabel.Text = 'Depth Range';
6160
6161            % Create LabelSpacingEditFieldLabel_2
6162            app.LabelSpacingEditFieldLabel_2 = uilabel(app.BathymetryTab);
6163            app.LabelSpacingEditFieldLabel_2.WordWrap = 'on';
6164            app.LabelSpacingEditFieldLabel_2.Enable = 'off';
6165            app.LabelSpacingEditFieldLabel_2.Tooltip = {'Adjust the distance↙
between the labels'; ''; 'Higher values reduce label crowding.'; ''; 'Unit: 1 =↙
1/72 inch'};
6166            app.LabelSpacingEditFieldLabel_2.Position = [377 25 58 22];
6167            app.LabelSpacingEditFieldLabel_2.Text = 'Spacing';
6168
6169            % Create LabelSpacingEditField
6170            app.LabelSpacingEditField = uieditfield(app.BathymetryTab,↙
'numeric');
6171            app.LabelSpacingEditField.Limits = [0 Inf];
6172            app.LabelSpacingEditField.ValueDisplayFormat = '%8.1f';
6173            app.LabelSpacingEditField.Enable = 'off';
6174            app.LabelSpacingEditField.Tooltip = {''};
6175            app.LabelSpacingEditField.Position = [426 28 47 16];
6176            app.LabelSpacingEditField.Value = 1000;
6177
6178            % Create WidthEditFieldLabel
6179            app.WidthEditFieldLabel = uilabel(app.BathymetryTab);
6180            app.WidthEditFieldLabel.Enable = 'off';
6181            app.WidthEditFieldLabel.Tooltip = {'Line thickness'};
6182            app.WidthEditFieldLabel.Position = [196 71 37 22];
6183            app.WidthEditFieldLabel.Text = 'Width';
6184
6185            % Create WidthEditField
6186            app.WidthEditField = uieditfield(app.BathymetryTab, 'numeric');
6187            app.WidthEditField.Limits = [0 Inf];
6188            app.WidthEditField.ValueDisplayFormat = '%3.1f';
6189            app.WidthEditField.Enable = 'off';
6190            app.WidthEditField.Tooltip = {''};
6191            app.WidthEditField.Position = [236 74 35 16];
6192            app.WidthEditField.Value = 0.1;
6193
6194            % Create LineLabel
6195            app.LineLabel = uilabel(app.BathymetryTab);
6196            app.LineLabel.FontWeight = 'bold';
6197            app.LineLabel.Enable = 'off';
6198            app.LineLabel.Position = [196 95 30 22];
6199            app.LineLabel.Text = 'Line';
6200
6201            % Create StyleDropDownLabel
6202            app.StyleDropDownLabel = uilabel(app.BathymetryTab);
6203            app.StyleDropDownLabel.Enable = 'off';
6204            app.StyleDropDownLabel.Position = [196 49 32 22];
6205            app.StyleDropDownLabel.Text = 'Style';
6206
6207            % Create StyleDropDown
6208            app.StyleDropDown = uidropdown(app.BathymetryTab);
6209            app.StyleDropDown.Items = {'Solid', 'Dashed', 'Dotted', 'Dash-↙
dotted'};
```

```
6210             app.StyleDropDown.Enable = 'off';
6211             app.StyleDropDown.Position = [236 52 104 16];
6212             app.StyleDropDown.Value = 'Solid';
6213 
6214             % Create ColorDropDownLabel
6215             app.ColorDropDownLabel = uilabel(app.BathymetryTab);
6216             app.ColorDropDownLabel.Enable = 'off';
6217             app.ColorDropDownLabel.Position = [196 25 35 22];
6218             app.ColorDropDownLabel.Text = 'Color';
6219 
6220             % Create ColorDropDown
6221             app.ColorDropDown = uidropdown(app.BathymetryTab);
6222             app.ColorDropDown.Items = {'Black', 'Dark gray', 'Medium gray',↙
'Light gray', 'Red', 'Green', 'Blue', 'Yellow', 'Cyan', 'Magenta', 'White'};
6223             app.ColorDropDown.Enable = 'off';
6224             app.ColorDropDown.Position = [236 28 104 16];
6225             app.ColorDropDown.Value = 'Dark gray';
6226 
6227             % Create AddLabelsCheckBox_2
6228             app.AddLabelsCheckBox_2 = uicheckbox(app.BathymetryTab);
6229             app.AddLabelsCheckBox_2.ValueChangedFcn = createCallbackFcn(app,↙
@AddLabelsCheckBox_2ValueChanged2, true);
6230             app.AddLabelsCheckBox_2.Enable = 'off';
6231             app.AddLabelsCheckBox_2.Tooltip = {'Unit: meters'};
6232             app.AddLabelsCheckBox_2.Text = ' Add Labels';
6233             app.AddLabelsCheckBox_2.WordWrap = 'on';
6234             app.AddLabelsCheckBox_2.FontWeight = 'bold';
6235             app.AddLabelsCheckBox_2.Position = [377 95 93 22];
6236 
6237             % Create TextLabelSize
6238             app.TextLabelSize = uieditfield(app.BathymetryTab, 'numeric');
6239             app.TextLabelSize.Limits = [0 30];
6240             app.TextLabelSize.ValueDisplayFormat = '%8.0f';
6241             app.TextLabelSize.Enable = 'off';
6242             app.TextLabelSize.Tooltip = {''};
6243             app.TextLabelSize.Position = [426 74 30 16];
6244             app.TextLabelSize.Value = 8;
6245 
6246             % Create SpacingEditField_4Label_4
6247             app.SpacingEditField_4Label_4 = uilabel(app.BathymetryTab);
6248             app.SpacingEditField_4Label_4.WordWrap = 'on';
6249             app.SpacingEditField_4Label_4.Enable = 'off';
6250             app.SpacingEditField_4Label_4.Position = [377 71 40 22];
6251             app.SpacingEditField_4Label_4.Text = 'Size';
6252 
6253             % Create LabelSpacingEditFieldLabel
6254             app.LabelSpacingEditFieldLabel = uilabel(app.BathymetryTab);
6255             app.LabelSpacingEditFieldLabel.Enable = 'off';
6256             app.LabelSpacingEditFieldLabel.Tooltip = {'Add labels at specified↙
intervals.'; ''; 'Suggestion: Set at intervals divisible by the depth range↙
interval'};
6257             app.LabelSpacingEditFieldLabel.Position = [377 49 45 22];
6258             app.LabelSpacingEditFieldLabel.Text = 'Interval';
6259 
6260             % Create IntervalEditField_3
6261             app.IntervalEditField_3 = uieditfield(app.BathymetryTab,↙
'numeric');
6262             app.IntervalEditField_3.Limits = [0.01 Inf];
```

```
6263            app.IntervalEditField_3.ValueDisplayFormat = '%8.1f';
6264            app.IntervalEditField_3.Enable = 'off';
6265            app.IntervalEditField_3.Tooltip = {''};
6266            app.IntervalEditField_3.Position = [426 52 47 16];
6267            app.IntervalEditField_3.Value = 500;
6268
6269            % Create WaveHeightTab
6270            app.WaveHeightTab = uitab(app.TabGroup2);
6271            app.WaveHeightTab.AutoResizeChildren = 'off';
6272            app.WaveHeightTab.Title = 'Wave Height';
6273
6274            % Create MinimumdepthEditField_2
6275            app.MinimumdepthEditField_2 = uieditfield(app.WaveHeightTab,↙
'numeric');
6276            app.MinimumdepthEditField_2.ValueDisplayFormat = '%8.1f';
6277            app.MinimumdepthEditField_2.Tooltip = {'Minimum value of the wave↙
height'};
6278            app.MinimumdepthEditField_2.Position = [107 74 45 16];
6279
6280            % Create MaximumdepthEditField_2
6281            app.MaximumdepthEditField_2 = uieditfield(app.WaveHeightTab,↙
'numeric');
6282            app.MaximumdepthEditField_2.Limits = [1 Inf];
6283            app.MaximumdepthEditField_2.ValueDisplayFormat = '%8.1f';
6284            app.MaximumdepthEditField_2.Tooltip = {'Maximum value of the wave↙
height'};
6285            app.MaximumdepthEditField_2.Position = [107 29 45 16];
6286            app.MaximumdepthEditField_2.Value = 1;
6287
6288            % Create LineintervalEditField_2
6289            app.LineintervalEditField_2 = uieditfield(app.WaveHeightTab,↙
'numeric');
6290            app.LineintervalEditField_2.Limits = [0 Inf];
6291            app.LineintervalEditField_2.ValueDisplayFormat = '%8.2f';
6292            app.LineintervalEditField_2.HorizontalAlignment = 'center';
6293            app.LineintervalEditField_2.Tooltip = {'Interval between adjacent↙
contours'};
6294            app.LineintervalEditField_2.Position = [107 51 34 16];
6295            app.LineintervalEditField_2.Value = 0.5;
6296
6297            % Create MaximumdepthEditFieldLabel_2
6298            app.MaximumdepthEditFieldLabel_2 = uilabel(app.WaveHeightTab);
6299            app.MaximumdepthEditFieldLabel_2.Position = [48 25 53 22];
6300            app.MaximumdepthEditFieldLabel_2.Text = 'Maximum';
6301
6302            % Create MinimumdepthEditFieldLabel_2
6303            app.MinimumdepthEditFieldLabel_2 = uilabel(app.WaveHeightTab);
6304            app.MinimumdepthEditFieldLabel_2.Position = [48 71 51 22];
6305            app.MinimumdepthEditFieldLabel_2.Text = 'Minimum';
6306
6307            % Create MaximumdepthEditFieldLabel_3
6308            app.MaximumdepthEditFieldLabel_3 = uilabel(app.WaveHeightTab);
6309            app.MaximumdepthEditFieldLabel_3.Position = [48 48 48 22];
6310            app.MaximumdepthEditFieldLabel_3.Text = 'Interval';
6311
6312            % Create ContourRangeLabel_2
6313            app.ContourRangeLabel_2 = uilabel(app.WaveHeightTab);
6314            app.ContourRangeLabel_2.FontWeight = 'bold';
```

```
6315            app.ContourRangeLabel_2.Position = [48 95 92 22];
6316            app.ContourRangeLabel_2.Text = 'Contour Range';
6317
6318            % Create LineLabel_2
6319            app.LineLabel_2 = uilabel(app.WaveHeightTab);
6320            app.LineLabel_2.FontWeight = 'bold';
6321            app.LineLabel_2.Position = [196 95 30 22];
6322            app.LineLabel_2.Text = 'Line';
6323
6324            % Create ThicknessEditField_2Label
6325            app.ThicknessEditField_2Label = uilabel(app.WaveHeightTab);
6326            app.ThicknessEditField_2Label.Tooltip = {'Line thickness'};
6327            app.ThicknessEditField_2Label.Position = [196 71 37 22];
6328            app.ThicknessEditField_2Label.Text = 'Width';
6329
6330            % Create ThicknessEditField_2
6331            app.ThicknessEditField_2 = uieditfield(app.WaveHeightTab,↙
'numeric');
6332            app.ThicknessEditField_2.Limits = [0 20];
6333            app.ThicknessEditField_2.ValueDisplayFormat = '%3.1f';
6334            app.ThicknessEditField_2.Position = [236 74 35 16];
6335            app.ThicknessEditField_2.Value = 0.1;
6336
6337            % Create ColorDropDown_2Label
6338            app.ColorDropDown_2Label = uilabel(app.WaveHeightTab);
6339            app.ColorDropDown_2Label.Position = [196 25 35 22];
6340            app.ColorDropDown_2Label.Text = 'Color';
6341
6342            % Create ColorDropDown_2
6343            app.ColorDropDown_2 = uidropdown(app.WaveHeightTab);
6344            app.ColorDropDown_2.Items = {'Black', 'Dark gray', 'Medium gray',↙
'Light gray', 'Red', 'Green', 'Blue', 'Yellow', 'Cyan', 'Magenta', 'White'};
6345            app.ColorDropDown_2.Position = [236 28 104 16];
6346            app.ColorDropDown_2.Value = 'Dark gray';
6347
6348            % Create StyleDropDown_2Label
6349            app.StyleDropDown_2Label = uilabel(app.WaveHeightTab);
6350            app.StyleDropDown_2Label.Position = [196 49 32 22];
6351            app.StyleDropDown_2Label.Text = 'Style';
6352
6353            % Create StyleDropDown_2
6354            app.StyleDropDown_2 = uidropdown(app.WaveHeightTab);
6355            app.StyleDropDown_2.Items = {'Solid', 'Dashed', 'Dotted', 'Dash-↙
dotted'};
6356            app.StyleDropDown_2.Position = [236 52 104 16];
6357            app.StyleDropDown_2.Value = 'Solid';
6358
6359            % Create AddLabelsCheckBox_3
6360            app.AddLabelsCheckBox_3 = uicheckbox(app.WaveHeightTab);
6361            app.AddLabelsCheckBox_3.ValueChangedFcn = createCallbackFcn(app,↙
@AddLabelsCheckBox_3ValueChanged2, true);
6362            app.AddLabelsCheckBox_3.Tooltip = {'Unit: meters'};
6363            app.AddLabelsCheckBox_3.Text = ' Add Labels';
6364            app.AddLabelsCheckBox_3.FontWeight = 'bold';
6365            app.AddLabelsCheckBox_3.Position = [377 95 90 22];
6366
6367            % Create TextLabelSize_2
6368            app.TextLabelSize_2 = uieditfield(app.WaveHeightTab, 'numeric');
```

```
6369            app.TextLabelSize_2.Limits = [0 Inf];
6370            app.TextLabelSize_2.ValueDisplayFormat = '%8.0f';
6371            app.TextLabelSize_2.Enable = 'off';
6372            app.TextLabelSize_2.Position = [426 74 30 16];
6373            app.TextLabelSize_2.Value = 8;
6374
6375            % Create SpacingEditField_4Label_5
6376            app.SpacingEditField_4Label_5 = uilabel(app.WaveHeightTab);
6377            app.SpacingEditField_4Label_5.WordWrap = 'on';
6378            app.SpacingEditField_4Label_5.Enable = 'off';
6379            app.SpacingEditField_4Label_5.Position = [377 71 40 22];
6380            app.SpacingEditField_4Label_5.Text = 'Size';
6381
6382            % Create LabelSpacingEditField_2Label_2
6383            app.LabelSpacingEditField_2Label_2 = uilabel(app.WaveHeightTab);
6384            app.LabelSpacingEditField_2Label_2.WordWrap = 'on';
6385            app.LabelSpacingEditField_2Label_2.Enable = 'off';
6386            app.LabelSpacingEditField_2Label_2.Tooltip = {'Adjust the distance↙
between the labels'; ''; 'Higher values reduce label crowding.'; ''; 'Unit: 1 =↙
1/72 inch'};
6387            app.LabelSpacingEditField_2Label_2.Position = [377 27 46 18];
6388            app.LabelSpacingEditField_2Label_2.Text = 'Spacing';
6389
6390            % Create LabelSpacingEditField_2
6391            app.LabelSpacingEditField_2 = uieditfield(app.WaveHeightTab, ↙
'numeric');
6392            app.LabelSpacingEditField_2.Limits = [0 Inf];
6393            app.LabelSpacingEditField_2.ValueDisplayFormat = '%8.1f';
6394            app.LabelSpacingEditField_2.Enable = 'off';
6395            app.LabelSpacingEditField_2.Tooltip = {''};
6396            app.LabelSpacingEditField_2.Position = [426 28 47 16];
6397            app.LabelSpacingEditField_2.Value = 1000;
6398
6399            % Create LabelSpacingEditField_2Label
6400            app.LabelSpacingEditField_2Label = uilabel(app.WaveHeightTab);
6401            app.LabelSpacingEditField_2Label.Enable = 'off';
6402            app.LabelSpacingEditField_2Label.Tooltip = {'Add labels at↙
specified intervals'; ''; 'Suggestion: Set at intervals divisible by the contour↙
range interval'};
6403            app.LabelSpacingEditField_2Label.Position = [377 49 45 22];
6404            app.LabelSpacingEditField_2Label.Text = 'Interval';
6405
6406            % Create wh_interval
6407            app.wh_interval = uieditfield(app.WaveHeightTab, 'numeric');
6408            app.wh_interval.Limits = [0.01 Inf];
6409            app.wh_interval.ValueDisplayFormat = '%8.1f';
6410            app.wh_interval.Enable = 'off';
6411            app.wh_interval.Tooltip = {''};
6412            app.wh_interval.Position = [426 52 47 16];
6413            app.wh_interval.Value = 1;
6414
6415            % Create ArrivalTimeTab
6416            app.ArrivalTimeTab = uitab(app.TabGroup2);
6417            app.ArrivalTimeTab.AutoResizeChildren = 'off';
6418            app.ArrivalTimeTab.Title = 'Arrival Time';
6419            app.ArrivalTimeTab.ForegroundColor = [0.9412 0.9412 0.9412];
6420
6421            % Create TimeFileButton
```

```
6422             app.TimeFileButton = uibutton(app.ArrivalTimeTab, 'push');
6423             app.TimeFileButton.ButtonPushedFcn = createCallbackFcn(app,
@TimeFileButtonPushed2, true);
6424             app.TimeFileButton.Tooltip = {'Load the file.'; ''; 'File
extension must be .shp or .txt'; ''; 'Unit: automatically converted to minutes. ';
''; ''; 'The default threshold for detecting the first wave arrival time in FUNWAVE
is 0.001 meters.'};
6425             app.TimeFileButton.Position = [211 105 19 19];
6426             app.TimeFileButton.Text = '...';
6427
6428             % Create FileEditField_3Label
6429             app.FileEditField_3Label = uilabel(app.ArrivalTimeTab);
6430             app.FileEditField_3Label.FontWeight = 'bold';
6431             app.FileEditField_3Label.Position = [48 103 29 22];
6432             app.FileEditField_3Label.Text = 'File:';
6433
6434             % Create FileEditField_3
6435             app.FileEditField_3 = uieditfield(app.ArrivalTimeTab, 'text');
6436             app.FileEditField_3.Editable = 'off';
6437             app.FileEditField_3.Tooltip = {'Unit: automatically converted to
meters '; ''; 'Default threshold for the first wave arrival time in FUNWAVE is
0.001 m.'};
6438             app.FileEditField_3.Placeholder = 'time_xxxxx';
6439             app.FileEditField_3.Position = [80 106 125 16];
6440
6441             % Create ContourRangeLabel
6442             app.ContourRangeLabel = uilabel(app.ArrivalTimeTab);
6443             app.ContourRangeLabel.FontWeight = 'bold';
6444             app.ContourRangeLabel.Tooltip = {'Unit: minutes'};
6445             app.ContourRangeLabel.Position = [49 78 92 22];
6446             app.ContourRangeLabel.Text = 'Contour Range';
6447
6448             % Create MaximumdepthEditField_3
6449             app.MaximumdepthEditField_3 = uieditfield(app.ArrivalTimeTab,
'numeric');
6450             app.MaximumdepthEditField_3.Limits = [1 Inf];
6451             app.MaximumdepthEditField_3.ValueDisplayFormat = '%8.1f';
6452             app.MaximumdepthEditField_3.Tooltip = {'Unit: minutes'};
6453             app.MaximumdepthEditField_3.Position = [106 18 55 16];
6454             app.MaximumdepthEditField_3.Value = 1;
6455
6456             % Create MaximumdepthEditFieldLabel_9
6457             app.MaximumdepthEditFieldLabel_9 = uilabel(app.ArrivalTimeTab);
6458             app.MaximumdepthEditFieldLabel_9.Tooltip = {'Unit: minutes'};
6459             app.MaximumdepthEditFieldLabel_9.Position = [49 15 53 22];
6460             app.MaximumdepthEditFieldLabel_9.Text = 'Maximum';
6461
6462             % Create LineintervalEditField_3
6463             app.LineintervalEditField_3 = uieditfield(app.ArrivalTimeTab,
'numeric');
6464             app.LineintervalEditField_3.Limits = [0 Inf];
6465             app.LineintervalEditField_3.ValueDisplayFormat = '%8.1f';
6466             app.LineintervalEditField_3.Tooltip = {'Interval between adjacent
contours.'; ''; 'Unit: minutes'};
6467             app.LineintervalEditField_3.Position = [106 38 55 16];
6468             app.LineintervalEditField_3.Value = 100;
6469
6470             % Create MaximumdepthEditFieldLabel_10
```

```
6471            app.MaximumdepthEditFieldLabel_10 = uilabel(app.ArrivalTimeTab);
6472            app.MaximumdepthEditFieldLabel_10.Tooltip = {'Unit: minutes'};
6473            app.MaximumdepthEditFieldLabel_10.Position = [49 35 48 22];
6474            app.MaximumdepthEditFieldLabel_10.Text = 'Interval';
6475
6476            % Create MinimumdepthEditField_3
6477            app.MinimumdepthEditField_3 = uieditfield(app.ArrivalTimeTab, ↙
'numeric');
6478            app.MinimumdepthEditField_3.Limits = [0 Inf];
6479            app.MinimumdepthEditField_3.ValueDisplayFormat = '%8.1f';
6480            app.MinimumdepthEditField_3.Tooltip = {'Unit: minutes'};
6481            app.MinimumdepthEditField_3.Position = [106 59 55 16];
6482            app.MinimumdepthEditField_3.Value = 10;
6483
6484            % Create MinimumdepthEditFieldLabel_5
6485            app.MinimumdepthEditFieldLabel_5 = uilabel(app.ArrivalTimeTab);
6486            app.MinimumdepthEditFieldLabel_5.Tooltip = {'Unit: minutes'};
6487            app.MinimumdepthEditFieldLabel_5.Position = [49 56 51 22];
6488            app.MinimumdepthEditFieldLabel_5.Text = 'Minimum';
6489
6490            % Create LabelSpacingEditField_2Label_4
6491            app.LabelSpacingEditField_2Label_4 = uilabel(app.ArrivalTimeTab);
6492            app.LabelSpacingEditField_2Label_4.Enable = 'off';
6493            app.LabelSpacingEditField_2Label_4.Tooltip = {'Add labels at ↙
specified intervals.'; ''; 'Suggestion: Set at intervals divisible by the contour ↙
range interval'; ''; 'Unit: meters'};
6494            app.LabelSpacingEditField_2Label_4.Position = [392 35 45 22];
6495            app.LabelSpacingEditField_2Label_4.Text = 'Interval';
6496
6497            % Create IntervalEditField_7
6498            app.IntervalEditField_7 = uieditfield(app.ArrivalTimeTab, ↙
'numeric');
6499            app.IntervalEditField_7.Limits = [0.01 Inf];
6500            app.IntervalEditField_7.ValueDisplayFormat = '%8.1f';
6501            app.IntervalEditField_7.Enable = 'off';
6502            app.IntervalEditField_7.Tooltip = {''};
6503            app.IntervalEditField_7.Position = [441 38 41 16];
6504            app.IntervalEditField_7.Value = 1;
6505
6506            % Create TextLabelSize_3
6507            app.TextLabelSize_3 = uieditfield(app.ArrivalTimeTab, 'numeric');
6508            app.TextLabelSize_3.Limits = [0 Inf];
6509            app.TextLabelSize_3.ValueDisplayFormat = '%8.0f';
6510            app.TextLabelSize_3.Enable = 'off';
6511            app.TextLabelSize_3.Tooltip = {''};
6512            app.TextLabelSize_3.Position = [441 59 30 16];
6513            app.TextLabelSize_3.Value = 8;
6514
6515            % Create SpacingEditField_4Label_6
6516            app.SpacingEditField_4Label_6 = uilabel(app.ArrivalTimeTab);
6517            app.SpacingEditField_4Label_6.WordWrap = 'on';
6518            app.SpacingEditField_4Label_6.Enable = 'off';
6519            app.SpacingEditField_4Label_6.Tooltip = {'Unit: meters'};
6520            app.SpacingEditField_4Label_6.Position = [392 59 36 17];
6521            app.SpacingEditField_4Label_6.Text = 'Size';
6522
6523            % Create AddLabelsCheckBox_5
6524            app.AddLabelsCheckBox_5 = uicheckbox(app.ArrivalTimeTab);
```

```
6525             app.AddLabelsCheckBox_5.ValueChangedFcn = createCallbackFcn(app, ↙
@AddLabelsCheckBox_5ValueChanged2, true);
6526             app.AddLabelsCheckBox_5.Tooltip = {'Unit: meters'};
6527             app.AddLabelsCheckBox_5.Text = 'Add Labels';
6528             app.AddLabelsCheckBox_5.FontWeight = 'bold';
6529             app.AddLabelsCheckBox_5.Position = [392 78 86 22];
6530 
6531             % Create LineLabel_5
6532             app.LineLabel_5 = uilabel(app.ArrivalTimeTab);
6533             app.LineLabel_5.FontWeight = 'bold';
6534             app.LineLabel_5.Position = [201 78 30 22];
6535             app.LineLabel_5.Text = 'Line';
6536 
6537             % Create ThicknessEditField_3
6538             app.ThicknessEditField_3 = uieditfield(app.ArrivalTimeTab, ↙
'numeric');
6539             app.ThicknessEditField_3.Limits = [0 20];
6540             app.ThicknessEditField_3.ValueDisplayFormat = '%3.1f';
6541             app.ThicknessEditField_3.Position = [239 59 35 16];
6542             app.ThicknessEditField_3.Value = 0.1;
6543 
6544             % Create ThicknessEditField_2Label_2
6545             app.ThicknessEditField_2Label_2 = uilabel(app.ArrivalTimeTab);
6546             app.ThicknessEditField_2Label_2.Position = [201 56 39 22];
6547             app.ThicknessEditField_2Label_2.Text = 'Width';
6548 
6549             % Create ColorDropDown_6Label
6550             app.ColorDropDown_6Label = uilabel(app.ArrivalTimeTab);
6551             app.ColorDropDown_6Label.Position = [200 15 35 22];
6552             app.ColorDropDown_6Label.Text = 'Color';
6553 
6554             % Create ColorDropDown_6
6555             app.ColorDropDown_6 = uidropdown(app.ArrivalTimeTab);
6556             app.ColorDropDown_6.Items = {'Black', 'Dark gray', 'Medium gray',↙
'Light gray', 'Red', 'Green', 'Blue', 'Yellow', 'Cyan', 'Magenta', 'White'};
6557             app.ColorDropDown_6.Position = [239 18 104 16];
6558             app.ColorDropDown_6.Value = 'Dark gray';
6559 
6560             % Create StyleDropDown_4Label_2
6561             app.StyleDropDown_4Label_2 = uilabel(app.ArrivalTimeTab);
6562             app.StyleDropDown_4Label_2.Position = [201 35 32 22];
6563             app.StyleDropDown_4Label_2.Text = 'Style';
6564 
6565             % Create StyleDropDown_4
6566             app.StyleDropDown_4 = uidropdown(app.ArrivalTimeTab);
6567             app.StyleDropDown_4.Items = {'Solid', 'Dashed', 'Dotted', 'Dash-↙
dotted'};
6568             app.StyleDropDown_4.Position = [239 38 104 16];
6569             app.StyleDropDown_4.Value = 'Solid';
6570 
6571             % Create LabelSpacingEditField_4
6572             app.LabelSpacingEditField_4 = uieditfield(app.ArrivalTimeTab, ↙
'numeric');
6573             app.LabelSpacingEditField_4.Limits = [0 Inf];
6574             app.LabelSpacingEditField_4.ValueDisplayFormat = '%8.0f';
6575             app.LabelSpacingEditField_4.Enable = 'off';
6576             app.LabelSpacingEditField_4.Tooltip = {''};
6577             app.LabelSpacingEditField_4.Position = [441 18 41 16];
```

```
6578            app.LabelSpacingEditField_4.Value = 1000;
6579
6580            % Create LabelSpacingEditField_2Label_3
6581            app.LabelSpacingEditField_2Label_3 = uilabel(app.ArrivalTimeTab);
6582            app.LabelSpacingEditField_2Label_3.WordWrap = 'on';
6583            app.LabelSpacingEditField_2Label_3.Enable = 'off';
6584            app.LabelSpacingEditField_2Label_3.Tooltip = {'Adjust the distance↙
between the labels'; ''; 'Higher values reduce label crowding.'; ''; 'Unit: 1 =↙
1/72 inch'};
6585            app.LabelSpacingEditField_2Label_3.Position = [392 12 45 29];
6586            app.LabelSpacingEditField_2Label_3.Text = 'Spacing';
6587
6588            % Create GaugesTab
6589            app.GaugesTab = uitab(app.TabGroup2);
6590            app.GaugesTab.AutoResizeChildren = 'off';
6591            app.GaugesTab.Title = 'Gauges';
6592
6593            % Create Button_8
6594            app.Button_8 = uibutton(app.GaugesTab, 'push');
6595            app.Button_8.ButtonPushedFcn = createCallbackFcn(app,↙
@Button_8Pushed2, true);
6596            app.Button_8.Tooltip = {'Load the file'; ''; 'File extension must↙
be .shp or .txt (tab delimited)'};
6597            app.Button_8.Position = [209 96 19 19];
6598            app.Button_8.Text = '...';
6599
6600            % Create GaugeFileLabel
6601            app.GaugeFileLabel = uilabel(app.GaugesTab);
6602            app.GaugeFileLabel.FontWeight = 'bold';
6603            app.GaugeFileLabel.Tooltip = {'File extension must be .shp or .↙
txt'};
6604            app.GaugeFileLabel.Position = [38 94 26 22];
6605            app.GaugeFileLabel.Text = 'File';
6606
6607            % Create FileEditField
6608            app.FileEditField = uieditfield(app.GaugesTab, 'text');
6609            app.FileEditField.Editable = 'off';
6610            app.FileEditField.Tooltip = {''};
6611            app.FileEditField.Placeholder = '.shp or .txt';
6612            app.FileEditField.Position = [77 97 125 16];
6613
6614            % Create ColorDropDown_3Label
6615            app.ColorDropDown_3Label = uilabel(app.GaugesTab);
6616            app.ColorDropDown_3Label.Tooltip = {'Marker color'};
6617            app.ColorDropDown_3Label.Position = [38 24 35 22];
6618            app.ColorDropDown_3Label.Text = 'Color';
6619
6620            % Create ColorDropDown_3
6621            app.ColorDropDown_3 = uidropdown(app.GaugesTab);
6622            app.ColorDropDown_3.Items = {'Black', 'Dark gray', 'Medium gray',↙
'Light gray', 'Red', 'Green', 'Blue', 'Yellow', 'Cyan', 'Magenta', 'White'};
6623            app.ColorDropDown_3.Position = [77 27 103 16];
6624            app.ColorDropDown_3.Value = 'Black';
6625
6626            % Create CoastlinecolourDropDownLabel_3
6627            app.CoastlinecolourDropDownLabel_3 = uilabel(app.GaugesTab);
6628            app.CoastlinecolourDropDownLabel_3.Tooltip = {'Marker style'};
6629            app.CoastlinecolourDropDownLabel_3.Position = [38 46 32 22];
```

```
6630            app.CoastlinecolourDropDownLabel_3.Text = 'Style';
6631
6632            % Create CoastlinecolourDropDown_3
6633            app.CoastlinecolourDropDown_3 = uidropdown(app.GaugesTab);
6634            app.CoastlinecolourDropDown_3.Items = {'o', '+', '*', '.', 'x',↙
'-', '|', '^', 'v', '>', '<', 'diamond', 'hexagram', 'pentagram', 'square'};
6635            app.CoastlinecolourDropDown_3.Tooltip = {''};
6636            app.CoastlinecolourDropDown_3.Position = [79 49 92 16];
6637            app.CoastlinecolourDropDown_3.Value = 'o';
6638
6639            % Create SizeEditFieldLabel
6640            app.SizeEditFieldLabel = uilabel(app.GaugesTab);
6641            app.SizeEditFieldLabel.Tooltip = {'Marker size'};
6642            app.SizeEditFieldLabel.Position = [38 69 28 22];
6643            app.SizeEditFieldLabel.Text = 'Size';
6644
6645            % Create SizeEditField
6646            app.SizeEditField = uieditfield(app.GaugesTab, 'numeric');
6647            app.SizeEditField.Limits = [0 50];
6648            app.SizeEditField.ValueDisplayFormat = '%3.0f';
6649            app.SizeEditField.HorizontalAlignment = 'left';
6650            app.SizeEditField.Tooltip = {''};
6651            app.SizeEditField.Position = [79 72 40 16];
6652            app.SizeEditField.Value = 5;
6653
6654            % Create AddLabelsCheckBox_4
6655            app.AddLabelsCheckBox_4 = uicheckbox(app.GaugesTab);
6656            app.AddLabelsCheckBox_4.ValueChangedFcn = createCallbackFcn(app,↙
@AddLabelsCheckBox_4ValueChanged2, true);
6657            app.AddLabelsCheckBox_4.Tooltip = {'The label follows the sequence↙
of latitude-longitude combinations from the loaded file.'; ''; 'The labels↙
correspond to the numeric values in the sta_ files.'};
6658            app.AddLabelsCheckBox_4.Text = ' Add Labels';
6659            app.AddLabelsCheckBox_4.WordWrap = 'on';
6660            app.AddLabelsCheckBox_4.FontWeight = 'bold';
6661            app.AddLabelsCheckBox_4.Position = [267 91 96 28];
6662
6663            % Create AlignmentLabel
6664            app.AlignmentLabel = uilabel(app.GaugesTab);
6665            app.AlignmentLabel.Enable = 'off';
6666            app.AlignmentLabel.Tooltip = {'Horizontal and vertical alignments↙
relative to the marker location'};
6667            app.AlignmentLabel.Position = [267 24 59 22];
6668            app.AlignmentLabel.Text = 'Alignment';
6669
6670            % Create CoastlinecolourDropDown_4
6671            app.CoastlinecolourDropDown_4 = uidropdown(app.GaugesTab);
6672            app.CoastlinecolourDropDown_4.Items = {'Centre', 'Right', 'Left'};
6673            app.CoastlinecolourDropDown_4.Enable = 'off';
6674            app.CoastlinecolourDropDown_4.Tooltip = {''};
6675            app.CoastlinecolourDropDown_4.Position = [326 27 74 16];
6676            app.CoastlinecolourDropDown_4.Value = 'Right';
6677
6678            % Create CoastlinecolourDropDown_5
6679            app.CoastlinecolourDropDown_5 = uidropdown(app.GaugesTab);
6680            app.CoastlinecolourDropDown_5.Items = {'Centre', 'Top', 'Bottom'};
6681            app.CoastlinecolourDropDown_5.Enable = 'off';
6682            app.CoastlinecolourDropDown_5.Tooltip = {''};
```

```
6683            app.CoastlinecolourDropDown_5.Position = [406 27 74 16];
6684            app.CoastlinecolourDropDown_5.Value = 'Centre';
6685
6686            % Create SizeEditField_3Label
6687            app.SizeEditField_3Label = uilabel(app.GaugesTab);
6688            app.SizeEditField_3Label.Enable = 'off';
6689            app.SizeEditField_3Label.Position = [267 69 30 22];
6690            app.SizeEditField_3Label.Text = 'Size';
6691
6692            % Create SizeEditField_3
6693            app.SizeEditField_3 = uieditfield(app.GaugesTab, 'numeric');
6694            app.SizeEditField_3.Limits = [0 100];
6695            app.SizeEditField_3.ValueDisplayFormat = '%8.0f';
6696            app.SizeEditField_3.Enable = 'off';
6697            app.SizeEditField_3.Position = [326 72 40 16];
6698            app.SizeEditField_3.Value = 12;
6699
6700            % Create GLspacingLabel
6701            app.GLspacingLabel = uilabel(app.GaugesTab);
6702            app.GLspacingLabel.Enable = 'off';
6703            app.GLspacingLabel.Tooltip = {'Horizontal distance between the↙
point and the label.'; ''; ' A value of 0.5 means that a label for a point at 120°E↙
will be placed at 120.5°E.'; ' '; 'Unit: degrees / meters'};
6704            app.GLspacingLabel.Position = [267 46 50 22];
6705            app.GLspacingLabel.Text = 'Spacing';
6706
6707            % Create GLspacing
6708            app.GLspacing = uieditfield(app.GaugesTab, 'numeric');
6709            app.GLspacing.ValueDisplayFormat = '%8.3f';
6710            app.GLspacing.Enable = 'off';
6711            app.GLspacing.Tooltip = {''};
6712            app.GLspacing.Position = [326 49 40 16];
6713            app.GLspacing.Value = 1;
6714
6715            % Create Tab
6716            app.Tab = uitab(app.TabGroup2);
6717            app.Tab.AutoResizeChildren = 'off';
6718            app.Tab.Title = 'Tab';
6719
6720            % Create ButtonGroup_17
6721            app.ButtonGroup_17 = uibuttongroup(app.Panel_7);
6722            app.ButtonGroup_17.AutoResizeChildren = 'off';
6723            app.ButtonGroup_17.SelectionChangedFcn = createCallbackFcn(app,↙
@ButtonGroup_17SelectionChanged, true);
6724            app.ButtonGroup_17.BorderType = 'none';
6725            app.ButtonGroup_17.SizeChangedFcn = createCallbackFcn(app,↙
@ButtonGroup_17SizeChanged, true);
6726            app.ButtonGroup_17.Position = [1 126 518 37];
6727
6728            % Create BathymetryButton_2
6729            app.BathymetryButton_2 = uiradiobutton(app.ButtonGroup_17);
6730            app.BathymetryButton_2.Tooltip = {''};
6731            app.BathymetryButton_2.Text = 'Bathymetry';
6732            app.BathymetryButton_2.Position = [44 3 84 22];
6733
6734            % Create WaveHeightButton
6735            app.WaveHeightButton = uiradiobutton(app.ButtonGroup_17);
6736            app.WaveHeightButton.Tooltip = {'This is based on the uploaded↙
```

```
hmax/eta files'};
            app.WaveHeightButton.Text = 'Wave Height';
            app.WaveHeightButton.Position = [139 3 90 22];

            % Create ArrivalTimeButton
            app.ArrivalTimeButton = uiradiobutton(app.ButtonGroup_17);
            app.ArrivalTimeButton.Tooltip = {'Arrival time threshold set by
FUNWAVE: 0.001 m'};
            app.ArrivalTimeButton.Text = 'Arrival Time';
            app.ArrivalTimeButton.Position = [239 3 85 22];

            % Create NoneButton
            app.NoneButton = uiradiobutton(app.ButtonGroup_17);
            app.NoneButton.Text = 'None';
            app.NoneButton.Position = [338 3 51 22];
            app.NoneButton.Value = true;

            % Create GaugesCheckBox
            app.GaugesCheckBox = uicheckbox(app.Panel_7);
            app.GaugesCheckBox.ValueChangedFcn = createCallbackFcn(app,
@GaugesCheckBoxValueChanged, true);
            app.GaugesCheckBox.Tooltip = {''};
            app.GaugesCheckBox.Text = 'Gauges';
            app.GaugesCheckBox.Position = [419 129 64 22];

            % Create OverlayFeaturesLabel
            app.OverlayFeaturesLabel = uilabel(app.WaveHeightMapTab);
            app.OverlayFeaturesLabel.BackgroundColor = [0.9412 0.9412 0.9412];
            app.OverlayFeaturesLabel.FontSize = 15;
            app.OverlayFeaturesLabel.FontWeight = 'bold';
            app.OverlayFeaturesLabel.FontColor = [0.0314 0.3686 0.6];
            app.OverlayFeaturesLabel.Position = [38 471 143 22];
            app.OverlayFeaturesLabel.Text = '  Overlay Features  ';

            % Create Panel_5
            app.Panel_5 = uipanel(app.WaveHeightMapTab);
            app.Panel_5.AutoResizeChildren = 'off';
            app.Panel_5.Position = [11 494 520 157];

            % Create DivisionEditFieldLabel
            app.DivisionEditFieldLabel = uilabel(app.Panel_5);
            app.DivisionEditFieldLabel.FontSize = 11;
            app.DivisionEditFieldLabel.Enable = 'off';
            app.DivisionEditFieldLabel.Tooltip = {'Defines the color
distribution:'; ''; 'Low value:'; 'Distinct color boundaries'; ''; 'Higher value:';
'Smoother color transition'};
            app.DivisionEditFieldLabel.Position = [179 85 44 22];
            app.DivisionEditFieldLabel.Text = 'Division';

            % Create DivisionEditField
            app.DivisionEditField = uieditfield(app.Panel_5, 'numeric');
            app.DivisionEditField.Limits = [0 200];
            app.DivisionEditField.ValueDisplayFormat = '%3.0f';
            app.DivisionEditField.ValueChangedFcn = createCallbackFcn(app,
@DivisionEditFieldValueChanged, true);
            app.DivisionEditField.FontColor = [0.651 0.651 0.651];
            app.DivisionEditField.Enable = 'off';
            app.DivisionEditField.Tooltip = {''};
```

```
6789            app.DivisionEditField.Position = [223 88 30 16];
6790            app.DivisionEditField.Value = 100;
6791
6792            % Create FlipCheckBox_5
6793            app.FlipCheckBox_5 = uicheckbox(app.Panel_5);
6794            app.FlipCheckBox_5.ValueChangedFcn = createCallbackFcn(app,
@FlipCheckBox_5ValueChanged, true);
6795            app.FlipCheckBox_5.Enable = 'off';
6796            app.FlipCheckBox_5.Tooltip = {'Reverse the color sequence'};
6797            app.FlipCheckBox_5.Text = 'Flip';
6798            app.FlipCheckBox_5.FontSize = 11;
6799            app.FlipCheckBox_5.Position = [112 85 40 22];
6800
6801            % Create CoastlinecolourDropDown_8
6802            app.CoastlinecolourDropDown_8 = uidropdown(app.Panel_5);
6803            app.CoastlinecolourDropDown_8.Items = {'--- MATLAB default ----',
'autumn', 'bone', 'colorcube', 'cool', 'copper', 'flag', 'gray', 'hot', 'hsv',
'jet', 'parula', 'pink', 'prism', 'spring', 'summer', 'turbo', 'winter', '---
CBREWER 2 ---', '< sequential >', 'blue', 'blue - green', 'blue - purple', 'green -
blue', 'greens', 'grays', 'oranges', 'orange - red', 'purple - blue', 'purple -
blue - green', 'purple - red', 'purples', 'red - purple', 'reds', 'yellow - green',
'yellow - green - blue', 'yellow - orange - brown', 'yellow - orange - red', '<
divergent >', 'brown - teal', 'pink - light green', 'purple - green', 'purple -
orange', 'red - blue', 'red - gray', 'red - yellow - blue', 'red - yellow - green',
'spectral', '< qualitative >', 'accent', 'dark 2', 'paired', 'pastel 1', 'pastel
2', 'set 1', 'set 2', 'set 3'};
6804            app.CoastlinecolourDropDown_8.ValueChangedFcn = createCallbackFcn
(app, @CoastlinecolourDropDown_6ValueChanged, true);
6805            app.CoastlinecolourDropDown_8.Enable = 'off';
6806            app.CoastlinecolourDropDown_8.Tooltip = {''};
6807            app.CoastlinecolourDropDown_8.Position = [112 106 141 17];
6808            app.CoastlinecolourDropDown_8.Value = 'parula';
6809
6810            % Create CoastlinecolourDropDownLabel_8
6811            app.CoastlinecolourDropDownLabel_8 = uilabel(app.Panel_5);
6812            app.CoastlinecolourDropDownLabel_8.FontWeight = 'bold';
6813            app.CoastlinecolourDropDownLabel_8.Position = [26 104 62 22];
6814            app.CoastlinecolourDropDownLabel_8.Text = 'Colormap';
6815
6816            % Create hmaxLabel
6817            app.hmaxLabel = uilabel(app.Panel_5);
6818            app.hmaxLabel.WordWrap = 'on';
6819            app.hmaxLabel.FontWeight = 'bold';
6820            app.hmaxLabel.FontAngle = 'italic';
6821            app.hmaxLabel.Tooltip = {'Maximum wave height'};
6822            app.hmaxLabel.Position = [112 122 60 22];
6823            app.hmaxLabel.Text = ' hmax';
6824
6825            % Create MinEditField
6826            app.MinEditField = uieditfield(app.Panel_5, 'numeric');
6827            app.MinEditField.Limits = [0 Inf];
6828            app.MinEditField.ValueDisplayFormat = '%8.1f';
6829            app.MinEditField.ValueChangedFcn = createCallbackFcn(app,
@MinEditFieldValueChanged, true);
6830            app.MinEditField.FontColor = [0.651 0.651 0.651];
6831            app.MinEditField.Enable = 'off';
6832            app.MinEditField.Tooltip = {''};
6833            app.MinEditField.Position = [112 62 40 16];
```

```
6834
6835             % Create toEditField
6836             app.toEditField = uieditfield(app.Panel_5, 'numeric');
6837             app.toEditField.Limits = [0.1 Inf];
6838             app.toEditField.ValueDisplayFormat = '%.1f';
6839             app.toEditField.ValueChangedFcn = createCallbackFcn(app,
@toEditFieldValueChanged, true);
6840             app.toEditField.FontColor = [0.651 0.651 0.651];
6841             app.toEditField.Enable = 'off';
6842             app.toEditField.Tooltip = {''};
6843             app.toEditField.Position = [177 62 40 16];
6844             app.toEditField.Value = 5;
6845
6846             % Create CoastlinecolourDropDownLabel_10
6847             app.CoastlinecolourDropDownLabel_10 = uilabel(app.Panel_5);
6848             app.CoastlinecolourDropDownLabel_10.WordWrap = 'on';
6849             app.CoastlinecolourDropDownLabel_10.FontWeight = 'bold';
6850             app.CoastlinecolourDropDownLabel_10.Tooltip = {'Limit of the ave
height values to display on the map'};
6851             app.CoastlinecolourDropDownLabel_10.Position = [25 55 67 30];
6852             app.CoastlinecolourDropDownLabel_10.Text = 'Colorbar';
6853
6854             % Create toLabel
6855             app.toLabel = uilabel(app.Panel_5);
6856             app.toLabel.Position = [159 59 16 22];
6857             app.toLabel.Text = 'to';
6858
6859             % Create etaLabel
6860             app.etaLabel = uilabel(app.Panel_5);
6861             app.etaLabel.WordWrap = 'on';
6862             app.etaLabel.FontWeight = 'bold';
6863             app.etaLabel.FontAngle = 'italic';
6864             app.etaLabel.Tooltip = {'Sea surface displacement at time T'};
6865             app.etaLabel.Position = [311 122 57 22];
6866             app.etaLabel.Text = 'eta';
6867
6868             % Create CoastlinecolourDropDown_7
6869             app.CoastlinecolourDropDown_7 = uidropdown(app.Panel_5);
6870             app.CoastlinecolourDropDown_7.Items = {'--- MATLAB default ----',
'autumn', 'bone', 'colorcube', 'cool', 'copper', 'flag', 'gray', 'hot', 'hsv',
'jet', 'parula', 'pink', 'prism', 'spring', 'summer', 'turbo', 'winter', '---
CBREWER 2 ---', '< sequential >', 'blue', 'blue - green', 'blue - purple', 'green -
blue', 'greens', 'grays', 'oranges', 'orange - red', 'purple - blue', 'purple -
blue - green', 'purple - red', 'purples', 'red - purple', 'reds', 'yellow - green',
'yellow - green - blue', 'yellow - orange - brown', 'yellow - orange - red', '<
divergent >', 'brown - teal', 'pink - light green', 'purple - green', 'purple -
orange', 'red - blue', 'red - gray', 'red - yellow - blue', 'red - yellow - green',
'spectral', '< qualitative >', 'accent', 'dark 2', 'paired', 'pastel 1', 'pastel
2', 'set 1', 'set 2', 'set 3'};
6871             app.CoastlinecolourDropDown_7.ValueChangedFcn = createCallbackFcn
(app, @CoastlinecolourDropDown_7ValueChanged, true);
6872             app.CoastlinecolourDropDown_7.Enable = 'off';
6873             app.CoastlinecolourDropDown_7.Tooltip = {''};
6874             app.CoastlinecolourDropDown_7.Position = [311 106 141 17];
6875             app.CoastlinecolourDropDown_7.Value = 'red - blue';
6876
6877             % Create FlipCheckBox_2
6878             app.FlipCheckBox_2 = uicheckbox(app.Panel_5);
```

```
6879             app.FlipCheckBox_2.ValueChangedFcn = createCallbackFcn(app, @FlipCheckBox_2ValueChanged, true);
6880             app.FlipCheckBox_2.Enable = 'off';
6881             app.FlipCheckBox_2.Tooltip = {'Reverse the color sequence'};
6882             app.FlipCheckBox_2.Text = 'Flip';
6883             app.FlipCheckBox_2.FontSize = 11;
6884             app.FlipCheckBox_2.Position = [311 85 40 22];
6885             app.FlipCheckBox_2.Value = true;
6886 
6887             % Create ColorinterpolationEditField_2
6888             app.ColorinterpolationEditField_2 = uieditfield(app.Panel_5, 'numeric');
6889             app.ColorinterpolationEditField_2.Limits = [0 200];
6890             app.ColorinterpolationEditField_2.ValueDisplayFormat = '%3.0f';
6891             app.ColorinterpolationEditField_2.ValueChangedFcn = createCallbackFcn(app, @ColorinterpolationEditField_2ValueChanged, true);
6892             app.ColorinterpolationEditField_2.FontColor = [0.651 0.651 0.651];
6893             app.ColorinterpolationEditField_2.Enable = 'off';
6894             app.ColorinterpolationEditField_2.Tooltip = {''};
6895             app.ColorinterpolationEditField_2.Position = [422 88 30 16];
6896             app.ColorinterpolationEditField_2.Value = 100;
6897 
6898             % Create DivisionLabel
6899             app.DivisionLabel = uilabel(app.Panel_5);
6900             app.DivisionLabel.FontSize = 11;
6901             app.DivisionLabel.Enable = 'off';
6902             app.DivisionLabel.Tooltip = {'Defines the color distribution:'; '';'Low value:'; 'Distinct color boundaries'; ''; 'Higher value:'; 'Smoother color transition'};
6903             app.DivisionLabel.Position = [377 85 44 22];
6904             app.DivisionLabel.Text = 'Division';
6905 
6906             % Create MaxEditField_2
6907             app.MaxEditField_2 = uieditfield(app.Panel_5, 'numeric');
6908             app.MaxEditField_2.Limits = [0 Inf];
6909             app.MaxEditField_2.ValueDisplayFormat = '%.1f';
6910             app.MaxEditField_2.ValueChangedFcn = createCallbackFcn(app, @MaxEditField_2ValueChanged, true);
6911             app.MaxEditField_2.FontColor = [0.651 0.651 0.651];
6912             app.MaxEditField_2.Enable = 'off';
6913             app.MaxEditField_2.Tooltip = {'Suggestion: '; 'Set this value as the (+) of the min value to place the zero mark in the middle of the colorbar'};
6914             app.MaxEditField_2.Position = [376 62 40 16];
6915             app.MaxEditField_2.Value = 0.5;
6916 
6917             % Create MinEditField_2
6918             app.MinEditField_2 = uieditfield(app.Panel_5, 'numeric');
6919             app.MinEditField_2.ValueDisplayFormat = '%8.1f';
6920             app.MinEditField_2.ValueChangedFcn = createCallbackFcn(app, @MinEditField_2ValueChanged, true);
6921             app.MinEditField_2.FontColor = [0.651 0.651 0.651];
6922             app.MinEditField_2.Enable = 'off';
6923             app.MinEditField_2.Tooltip = {'Suggestion: '; 'Set this value as the (-) of the max value to place the zero mark in the middle of the colorbar'};
6924             app.MinEditField_2.Position = [311 62 41 16];
6925             app.MinEditField_2.Value = -0.5;
6926 
6927             % Create toLabel 2
```

```
6928            app.toLabel_2 = uilabel(app.Panel_5);
6929            app.toLabel_2.Position = [358 59 16 22];
6930            app.toLabel_2.Text = 'to';
6931
6932            % Create mLabel
6933            app.mLabel = uilabel(app.Panel_5);
6934            app.mLabel.Position = [223 59 25 22];
6935            app.mLabel.Text = 'm';
6936
6937            % Create mLabel_2
6938            app.mLabel_2 = uilabel(app.Panel_5);
6939            app.mLabel_2.Position = [423 59 25 22];
6940            app.mLabel_2.Text = 'm';
6941
6942            % Create CoastlinecolourDropDownLabel_9
6943            app.CoastlinecolourDropDownLabel_9 = uilabel(app.Panel_5);
6944            app.CoastlinecolourDropDownLabel_9.FontWeight = 'bold';
6945            app.CoastlinecolourDropDownLabel_9.Tooltip = {'Required input:';↙
'Bathymetry file'};
6946            app.CoastlinecolourDropDownLabel_9.Position = [307 28 69 22];
6947            app.CoastlinecolourDropDownLabel_9.Text = 'Land Color';
6948
6949            % Create HorizontalCheckBox
6950            app.HorizontalCheckBox = uicheckbox(app.Panel_5);
6951            app.HorizontalCheckBox.Text = 'Horizontal';
6952            app.HorizontalCheckBox.Position = [119 17 76 22];
6953
6954            % Create VerticalCheckBox
6955            app.VerticalCheckBox = uicheckbox(app.Panel_5);
6956            app.VerticalCheckBox.Text = 'Vertical';
6957            app.VerticalCheckBox.Position = [199 17 61 22];
6958
6959            % Create FlipBasemapLabel_2
6960            app.FlipBasemapLabel_2 = uilabel(app.Panel_5);
6961            app.FlipBasemapLabel_2.FontWeight = 'bold';
6962            app.FlipBasemapLabel_2.Tooltip = {'Flips the basemap.'; '';↙
'Horizontally (left-right)'; 'Vertically (up-side down)'};
6963            app.FlipBasemapLabel_2.Position = [25 17 87 22];
6964            app.FlipBasemapLabel_2.Text = 'Flip Basemap';
6965
6966            % Create CoastlinecolourDropDown_2
6967            app.CoastlinecolourDropDown_2 = uidropdown(app.Panel_5);
6968            app.CoastlinecolourDropDown_2.Items = {'Black', 'Dark gray',↙
'Medium gray', 'Light gray', 'Red', 'Green', 'Blue', 'Yellow', 'Cyan', 'Magenta',↙
'White'};
6969            app.CoastlinecolourDropDown_2.ValueChangedFcn = createCallbackFcn↙
(app, @CoastlinecolourDropDown_2ValueChanged, true);
6970            app.CoastlinecolourDropDown_2.Enable = 'off';
6971            app.CoastlinecolourDropDown_2.Tooltip = {''};
6972            app.CoastlinecolourDropDown_2.Position = [381 31 106 17];
6973            app.CoastlinecolourDropDown_2.Value = 'Dark gray';
6974
6975            % Create TextSizeLabel_2
6976            app.TextSizeLabel_2 = uilabel(app.Panel_5);
6977            app.TextSizeLabel_2.WordWrap = 'on';
6978            app.TextSizeLabel_2.FontWeight = 'bold';
6979            app.TextSizeLabel_2.Tooltip = {'Font size of the colorbar↙
labels'};
```

```
6980             app.TextSizeLabel_2.Position = [307 3 60 28];
6981             app.TextSizeLabel_2.Text = 'Text Size';
6982 
6983             % Create ColorbarTextSize
6984             app.ColorbarTextSize = uieditfield(app.Panel_5, 'numeric');
6985             app.ColorbarTextSize.Limits = [1 50];
6986             app.ColorbarTextSize.ValueDisplayFormat = '%2.0f';
6987             app.ColorbarTextSize.ValueChangedFcn = createCallbackFcn(app, ↙
@ColorbarTextSizeValueChanged, true);
6988             app.ColorbarTextSize.FontColor = [0.651 0.651 0.651];
6989             app.ColorbarTextSize.Enable = 'off';
6990             app.ColorbarTextSize.Tooltip = {''};
6991             app.ColorbarTextSize.Position = [381 9 29 16];
6992             app.ColorbarTextSize.Value = 11;
6993 
6994             % Create BasemapLabel
6995             app.BasemapLabel = uilabel(app.WaveHeightMapTab);
6996             app.BasemapLabel.BackgroundColor = [0.9412 0.9412 0.9412];
6997             app.BasemapLabel.FontSize = 15;
6998             app.BasemapLabel.FontWeight = 'bold';
6999             app.BasemapLabel.FontColor = [0.0314 0.3686 0.6];
7000             app.BasemapLabel.Tooltip = {'Modify the visuals of the imported↙
file/s (eta/hmax)'};
7001             app.BasemapLabel.Position = [38 639 89 22];
7002             app.BasemapLabel.Text = '  Basemap  ';
7003 
7004             % Create Panel_6
7005             app.Panel_6 = uipanel(app.WaveHeightMapTab);
7006             app.Panel_6.AutoResizeChildren = 'off';
7007             app.Panel_6.Position = [11 39 520 77];
7008 
7009             % Create pngCheckBox
7010             app.pngCheckBox = uicheckbox(app.Panel_6);
7011             app.pngCheckBox.Tooltip = {'Resolution is set to 300 DPI'};
7012             app.pngCheckBox.Text = '.png';
7013             app.pngCheckBox.Position = [191 12 46 22];
7014 
7015             % Create FileFormatLabel
7016             app.FileFormatLabel = uilabel(app.Panel_6);
7017             app.FileFormatLabel.Position = [69 12 66 22];
7018             app.FileFormatLabel.Text = 'File Format';
7019 
7020             % Create jpgCheckBox
7021             app.jpgCheckBox = uicheckbox(app.Panel_6);
7022             app.jpgCheckBox.Tooltip = {'Resolution is set to 300 DPI'};
7023             app.jpgCheckBox.Text = '.jpg';
7024             app.jpgCheckBox.Position = [139 12 42 22];
7025 
7026             % Create tifCheckBox_2
7027             app.tifCheckBox_2 = uicheckbox(app.Panel_6);
7028             app.tifCheckBox_2.ValueChangedFcn = createCallbackFcn(app, ↙
@tifCheckBox_2ValueChanged, true);
7029             app.tifCheckBox_2.Enable = 'off';
7030             app.tifCheckBox_2.Tooltip = {'Georeferenced raster file.'; '';↙
'Enabled when "Plot separately" is selected.'};
7031             app.tifCheckBox_2.Text = '.tif';
7032             app.tifCheckBox_2.Position = [246 12 35 22];
7033 
```

```
7034            % Create Button_2
7035            app.Button_2 = uibutton(app.Panel_6, 'push');
7036            app.Button_2.ButtonPushedFcn = createCallbackFcn(app,
@Button_2Pushed, true);
7037            app.Button_2.FontSize = 10;
7038            app.Button_2.FontAngle = 'italic';
7039            app.Button_2.Tooltip = {'Load the preferred directory.'; '';
'"OUTPUT_FILES/Figures"  folder will be created in the selected directory.'};
7040            app.Button_2.Position = [403 42 19 19];
7041            app.Button_2.Text = '...';
7042
7043            % Create OutputDirectoryLabel
7044            app.OutputDirectoryLabel = uilabel(app.Panel_6);
7045            app.OutputDirectoryLabel.Tooltip = {'Load the preferred
directory.'; ''; '''OUTPUT_FILES/Figures'' folder will be created in the selected
directory.'};
7046            app.OutputDirectoryLabel.Position = [40 40 94 22];
7047            app.OutputDirectoryLabel.Text = 'Output Directory';
7048
7049            % Create OutputDirectoryEditField
7050            app.OutputDirectoryEditField = uieditfield(app.Panel_6, 'text');
7051            app.OutputDirectoryEditField.Editable = 'off';
7052            app.OutputDirectoryEditField.FontColor = [0 0 1];
7053            app.OutputDirectoryEditField.Tooltip = {'Load the preferred
directory.'; ''; '''OUTPUT_FILES/Figures'' folder will be created in the selected
directory.'};
7054            app.OutputDirectoryEditField.Placeholder = 'Default: Desktop';
7055            app.OutputDirectoryEditField.Position = [139 43 257 16];
7056
7057            % Create FramerateEditField_2Label
7058            app.FramerateEditField_2Label = uilabel(app.Panel_6);
7059            app.FramerateEditField_2Label.WordWrap = 'on';
7060            app.FramerateEditField_2Label.Visible = 'off';
7061            app.FramerateEditField_2Label.Tooltip = {'Frames per second.'; '';
'Lower value = slower animation.'};
7062            app.FramerateEditField_2Label.Position = [346 10 68 27];
7063            app.FramerateEditField_2Label.Text = 'Frame rate';
7064
7065            % Create FramerateEditField_2
7066            app.FramerateEditField_2 = uieditfield(app.Panel_6, 'numeric');
7067            app.FramerateEditField_2.Limits = [0 100];
7068            app.FramerateEditField_2.ValueDisplayFormat = '%.0f';
7069            app.FramerateEditField_2.FontColor = [0.651 0.651 0.651];
7070            app.FramerateEditField_2.Visible = 'off';
7071            app.FramerateEditField_2.Tooltip = {''};
7072            app.FramerateEditField_2.Position = [412 15 30 17];
7073            app.FramerateEditField_2.Value = 5;
7074
7075            % Create mp4CheckBox
7076            app.mp4CheckBox = uicheckbox(app.Panel_6);
7077            app.mp4CheckBox.ValueChangedFcn = createCallbackFcn(app,
@mp4CheckBoxValueChanged, true);
7078            app.mp4CheckBox.Enable = 'off';
7079            app.mp4CheckBox.Tooltip = {'Create an animation.'; ''; 'File is
saved as "animation.mp4."'; ''; 'Enabled when "Plot separately" is selected.'};
7080            app.mp4CheckBox.Text = '.mp4';
7081            app.mp4CheckBox.Position = [290 12 49 22];
7082
```

```
7083            % Create SaveMapLabel
7084            app.SaveMapLabel = uilabel(app.WaveHeightMapTab);
7085            app.SaveMapLabel.BackgroundColor = [0.9412 0.9412 0.9412];
7086            app.SaveMapLabel.FontSize = 15;
7087            app.SaveMapLabel.FontWeight = 'bold';
7088            app.SaveMapLabel.FontColor = [0.0314 0.3686 0.6];
7089            app.SaveMapLabel.Position = [38 105 92 22];
7090            app.SaveMapLabel.Text = '  Save Map  ';
7091
7092            % Create GENERATEButton
7093            app.GENERATEButton = uibutton(app.WaveHeightMapTab, 'push');
7094            app.GENERATEButton.ButtonPushedFcn = createCallbackFcn(app,
@GENERATEButtonPushed, true);
7095            app.GENERATEButton.FontSize = 14;
7096            app.GENERATEButton.FontWeight = 'bold';
7097            app.GENERATEButton.Position = [179 6 183 25];
7098            app.GENERATEButton.Text = 'GENERATE';
7099
7100            % Create Panel_19
7101            app.Panel_19 = uipanel(app.WaveHeightMapTab);
7102            app.Panel_19.AutoResizeChildren = 'off';
7103            app.Panel_19.Position = [11 128 520 172];
7104
7105            % Create BoundaryLimitsLabel
7106            app.BoundaryLimitsLabel = uilabel(app.Panel_19);
7107            app.BoundaryLimitsLabel.FontWeight = 'bold';
7108            app.BoundaryLimitsLabel.Tooltip = {'When all textboxes are left
unchanged (all zeroes), the values are automatically set based on the extent of the
uploaded file and the input southwest corner coordinates'};
7109            app.BoundaryLimitsLabel.Position = [25 91 99 22];
7110            app.BoundaryLimitsLabel.Text = 'Boundary Limits';
7111
7112            % Create PlotindegreesCheckBox_2
7113            app.PlotindegreesCheckBox_2 = uicheckbox(app.Panel_19);
7114            app.PlotindegreesCheckBox_2.ValueChangedFcn = createCallbackFcn
(app, @PlotindegreesCheckBox_2ValueChanged, true);
7115            app.PlotindegreesCheckBox_2.Tooltip = {'Change the label format
into geographic coordinates'};
7116            app.PlotindegreesCheckBox_2.Text = 'Plot in degrees';
7117            app.PlotindegreesCheckBox_2.FontSize = 11;
7118            app.PlotindegreesCheckBox_2.Position = [403 56 96 22];
7119
7120            % Create EastEditFieldLabel
7121            app.EastEditFieldLabel = uilabel(app.Panel_19);
7122            app.EastEditFieldLabel.Position = [331 62 29 22];
7123            app.EastEditFieldLabel.Text = 'East';
7124
7125            % Create EastEditField
7126            app.EastEditField = uieditfield(app.Panel_19, 'numeric');
7127            app.EastEditField.ValueDisplayFormat = '%8.4f';
7128            app.EastEditField.ValueChangedFcn = createCallbackFcn(app,
@EastEditFieldValueChanged, true);
7129            app.EastEditField.FontColor = [0.651 0.651 0.651];
7130            app.EastEditField.Position = [265 65 60 16];
7131
7132            % Create WestEditFieldLabel
7133            app.WestEditFieldLabel = uilabel(app.Panel_19);
7134            app.WestEditFieldLabel.HorizontalAlignment = 'right';
```

```
7135            app.WestEditFieldLabel.Position = [87 62 32 22];
7136            app.WestEditFieldLabel.Text = 'West';
7137
7138            % Create WestEditField
7139            app.WestEditField = uieditfield(app.Panel_19, 'numeric');
7140            app.WestEditField.ValueDisplayFormat = '%8.4f';
7141            app.WestEditField.ValueChangedFcn = createCallbackFcn(app,
@WestEditFieldValueChanged, true);
7142            app.WestEditField.FontColor = [0.651 0.651 0.651];
7143            app.WestEditField.Position = [125 65 60 16];
7144
7145            % Create NorthEditFieldLabel
7146            app.NorthEditFieldLabel = uilabel(app.Panel_19);
7147            app.NorthEditFieldLabel.HorizontalAlignment = 'right';
7148            app.NorthEditFieldLabel.Position = [206 91 35 22];
7149            app.NorthEditFieldLabel.Text = 'North';
7150
7151            % Create NorthEditField
7152            app.NorthEditField = uieditfield(app.Panel_19, 'numeric');
7153            app.NorthEditField.ValueDisplayFormat = '%8.4f';
7154            app.NorthEditField.ValueChangedFcn = createCallbackFcn(app,
@NorthEditFieldValueChanged, true);
7155            app.NorthEditField.FontColor = [0.651 0.651 0.651];
7156            app.NorthEditField.Position = [195 77 60 16];
7157
7158            % Create SouthEditFieldLabel
7159            app.SouthEditFieldLabel = uilabel(app.Panel_19);
7160            app.SouthEditFieldLabel.HorizontalAlignment = 'right';
7161            app.SouthEditFieldLabel.Position = [205 34 37 22];
7162            app.SouthEditFieldLabel.Text = 'South';
7163
7164            % Create SouthEditField
7165            app.SouthEditField = uieditfield(app.Panel_19, 'numeric');
7166            app.SouthEditField.ValueDisplayFormat = '%8.4f';
7167            app.SouthEditField.ValueChangedFcn = createCallbackFcn(app,
@SouthEditFieldValueChanged, true);
7168            app.SouthEditField.FontColor = [0.651 0.651 0.651];
7169            app.SouthEditField.Position = [195 53 60 16];
7170
7171            % Create Height_2
7172            app.Height_2 = uieditfield(app.Panel_19, 'numeric');
7173            app.Height_2.Limits = [1 Inf];
7174            app.Height_2.ValueDisplayFormat = '%2.0f';
7175            app.Height_2.ValueChangedFcn = createCallbackFcn(app,
@Height_2ValueChanged, true);
7176            app.Height_2.FontColor = [0.651 0.651 0.651];
7177            app.Height_2.Enable = 'off';
7178            app.Height_2.Tooltip = {''};
7179            app.Height_2.Position = [330 12 40 16];
7180            app.Height_2.Value = 11;
7181
7182            % Create ThicknessLabel_8
7183            app.ThicknessLabel_8 = uilabel(app.Panel_19);
7184            app.ThicknessLabel_8.HorizontalAlignment = 'right';
7185            app.ThicknessLabel_8.Position = [283 9 41 22];
7186            app.ThicknessLabel_8.Text = 'Height';
7187
7188            % Create AutoSetCheckBox 2
```

```
7189            app.AutoSetCheckBox_2 = uicheckbox(app.Panel_19);
7190            app.AutoSetCheckBox_2.ValueChangedFcn = createCallbackFcn(app, ↙
@AutoSetCheckBox_2ValueChanged, true);
7191            app.AutoSetCheckBox_2.Text = ' Auto Set';
7192            app.AutoSetCheckBox_2.FontSize = 11;
7193            app.AutoSetCheckBox_2.Position = [102 9 68 22];
7194            app.AutoSetCheckBox_2.Value = true;
7195
7196            % Create Width_2
7197            app.Width_2 = uieditfield(app.Panel_19, 'numeric');
7198            app.Width_2.Limits = [1 Inf];
7199            app.Width_2.ValueDisplayFormat = '%2.0f';
7200            app.Width_2.ValueChangedFcn = createCallbackFcn(app, ↙
@Width_2ValueChanged, true);
7201            app.Width_2.FontColor = [0.651 0.651 0.651];
7202            app.Width_2.Enable = 'off';
7203            app.Width_2.Tooltip = {''};
7204            app.Width_2.Position = [225 12 41 16];
7205            app.Width_2.Value = 8;
7206
7207            % Create ThicknessLabel_9
7208            app.ThicknessLabel_9 = uilabel(app.Panel_19);
7209            app.ThicknessLabel_9.Tooltip = {'Unit: inches'};
7210            app.ThicknessLabel_9.Position = [188 9 37 22];
7211            app.ThicknessLabel_9.Text = 'Width';
7212
7213            % Create FigureSizeLabel_2
7214            app.FigureSizeLabel_2 = uilabel(app.Panel_19);
7215            app.FigureSizeLabel_2.FontWeight = 'bold';
7216            app.FigureSizeLabel_2.Position = [25 9 69 22];
7217            app.FigureSizeLabel_2.Text = 'Figure Size';
7218
7219            % Create CloseFiguresButton_4
7220            app.CloseFiguresButton_4 = uibutton(app.Panel_19, 'push');
7221            app.CloseFiguresButton_4.ButtonPushedFcn = createCallbackFcn(app, ↙
@CloseFiguresButton_4Pushed2, true);
7222            app.CloseFiguresButton_4.Tooltip = {''};
7223            app.CloseFiguresButton_4.Position = [403 10 89 20];
7224            app.CloseFiguresButton_4.Text = 'Close Figures';
7225
7226            % Create SettoDefaultButton_2
7227            app.SettoDefaultButton_2 = uibutton(app.Panel_19, 'push');
7228            app.SettoDefaultButton_2.ButtonPushedFcn = createCallbackFcn(app, ↙
@SettoDefaultButton_2Pushed, true);
7229            app.SettoDefaultButton_2.Tooltip = {'Use the boundary limits of ↙
the eta/hmax file'};
7230            app.SettoDefaultButton_2.Position = [403 76 84 20];
7231            app.SettoDefaultButton_2.Text = 'Set to Default';
7232
7233            % Create ButtonGroup_11
7234            app.ButtonGroup_11 = uibuttongroup(app.Panel_19);
7235            app.ButtonGroup_11.AutoResizeChildren = 'off';
7236            app.ButtonGroup_11.SelectionChangedFcn = createCallbackFcn(app, ↙
@ButtonGroup_11SelectionChanged, true);
7237            app.ButtonGroup_11.Position = [106 126 311 30];
7238
7239            % Create PlotalldatainonefigureButton
7240            app.PlotalldatainonefigureButton = uiradiobutton(app. ↙
```

```
ButtonGroup_11);
            app.PlotalldatainonefigureButton.Tooltip = {'All maps are displayed as subplots in one window'};
            app.PlotalldatainonefigureButton.Text = 'Plot all data in one figure';
            app.PlotalldatainonefigureButton.FontWeight = 'bold';
            app.PlotalldatainonefigureButton.Position = [13 4 164 22];
            app.PlotalldatainonefigureButton.Value = true;

            % Create PlotseparatelyButton_2
            app.PlotseparatelyButton_2 = uiradiobutton(app.ButtonGroup_11);
            app.PlotseparatelyButton_2.Tooltip = {'One map per window'};
            app.PlotseparatelyButton_2.Text = 'Plot separately';
            app.PlotseparatelyButton_2.FontWeight = 'bold';
            app.PlotseparatelyButton_2.Position = [193 4 108 22];

            % Create GeneralLayoutLabel_3
            app.GeneralLayoutLabel_3 = uilabel(app.WaveHeightMapTab);
            app.GeneralLayoutLabel_3.BackgroundColor = [0.9412 0.9412 0.9412];
            app.GeneralLayoutLabel_3.FontSize = 15;
            app.GeneralLayoutLabel_3.FontWeight = 'bold';
            app.GeneralLayoutLabel_3.FontColor = [0.0314 0.3686 0.6];
            app.GeneralLayoutLabel_3.Position = [38 290 131 22];
            app.GeneralLayoutLabel_3.Text = '  General Layout  ';

            % Create GaugeRecordsTab
            app.GaugeRecordsTab = uitab(app.TabGroup);
            app.GaugeRecordsTab.AutoResizeChildren = 'off';
            app.GaugeRecordsTab.Tooltip = {'When ''3 & 4'' is used for Y data, default values:'; ''; 'Column 3: Dashed-dotted lines'; 'Column 4: Dotted lines'};
            app.GaugeRecordsTab.SizeChangedFcn = createCallbackFcn(app, @GaugeRecordsTabSizeChanged, true);
            app.GaugeRecordsTab.Title = 'Gauge Records';
            app.GaugeRecordsTab.Scrollable = 'on';

            % Create Panel_10
            app.Panel_10 = uipanel(app.GaugeRecordsTab);
            app.Panel_10.AutoResizeChildren = 'off';
            app.Panel_10.Interruptible = 'off';
            app.Panel_10.Position = [11 129 520 562];

            % Create PlotPropertiesLabel
            app.PlotPropertiesLabel = uilabel(app.Panel_10);
            app.PlotPropertiesLabel.FontWeight = 'bold';
            app.PlotPropertiesLabel.Position = [50 179 91 22];
            app.PlotPropertiesLabel.Text = 'Plot Properties';

            % Create ColorDropDownLabel_2
            app.ColorDropDownLabel_2 = uilabel(app.Panel_10);
            app.ColorDropDownLabel_2.Tooltip = {''; 'When column ''3,4,5'''' is selected for Y axis, the default colors are: '; ''; '3 - purple'; '4 - green'; '5 - black'};
            app.ColorDropDownLabel_2.Position = [50 85 35 22];
            app.ColorDropDownLabel_2.Text = 'Color';

            % Create GaugeLineColor
            app.GaugeLineColor = uidropdown(app.Panel_10);
            app.GaugeLineColor.Items = {'Black', 'Dark gray', 'Medium gray',
```

```
'Light gray', 'Red', 'Green', 'Blue', 'Yellow', 'Cyan', 'Magenta'};
            app.GaugeLineColor.Visible = 'off';
            app.GaugeLineColor.Position = [108 87 89 17];
            app.GaugeLineColor.Value = 'Black';

            % Create LineStyleLabel
            app.LineStyleLabel = uilabel(app.Panel_10);
            app.LineStyleLabel.Tooltip = {'Line style of the plotted data.';↙
''; 'When column ''3,4,5'''' is selected for Y axis, the default styles are:'; '';↙
'3, 4 - dashed lines'; '5 - solid line'};
            app.LineStyleLabel.Position = [50 131 28 22];
            app.LineStyleLabel.Text = 'Line';

            % Create GaugeLineStyle
            app.GaugeLineStyle = uidropdown(app.Panel_10);
            app.GaugeLineStyle.Items = {'Solid', 'Dashed', 'Dotted', 'Dash-↙
dotted'};
            app.GaugeLineStyle.Tooltip = {''};
            app.GaugeLineStyle.Position = [107 134 91 17];
            app.GaugeLineStyle.Value = 'Solid';

            % Create ButtonGroup_10
            app.ButtonGroup_10 = uibuttongroup(app.Panel_10);
            app.ButtonGroup_10.AutoResizeChildren = 'off';
            app.ButtonGroup_10.SelectionChangedFcn = createCallbackFcn(app, ↙
@ButtonGroup_10SelectionChanged, true);
            app.ButtonGroup_10.FontWeight = 'bold';
            app.ButtonGroup_10.Position = [104 510 311 30];

            % Create PlotalldatainonegraphButton
            app.PlotalldatainonegraphButton = uiradiobutton(app.↙
ButtonGroup_10);
            app.PlotalldatainonegraphButton.Tooltip = {'Alll data are↙
summarized in one graph'};
            app.PlotalldatainonegraphButton.Text = 'Plot all data in one↙
graph';
            app.PlotalldatainonegraphButton.FontWeight = 'bold';
            app.PlotalldatainonegraphButton.Position = [13 4 165 22];
            app.PlotalldatainonegraphButton.Value = true;

            % Create PlotseparatelyButton_4
            app.PlotseparatelyButton_4 = uiradiobutton(app.ButtonGroup_10);
            app.PlotseparatelyButton_4.Tooltip = {'Data from each ''sta_''↙
file is plotted in separate windows'};
            app.PlotseparatelyButton_4.Text = 'Plot separately';
            app.PlotseparatelyButton_4.FontWeight = 'bold';
            app.PlotseparatelyButton_4.Position = [195 4 108 22];

            % Create DataLineWidth
            app.DataLineWidth = uieditfield(app.Panel_10, 'numeric');
            app.DataLineWidth.Limits = [0.1 Inf];
            app.DataLineWidth.ValueDisplayFormat = '%8.1f';
            app.DataLineWidth.ValueChangedFcn = createCallbackFcn(app, ↙
@DataLineWidthValueChanged, true);
            app.DataLineWidth.FontColor = [0.651 0.651 0.651];
            app.DataLineWidth.Tooltip = {''};
            app.DataLineWidth.Position = [107 111 29 16];
            app.DataLineWidth.Value = 1;
```

```
7340
7341            % Create WidthLabel
7342            app.WidthLabel = uilabel(app.Panel_10);
7343            app.WidthLabel.Position = [50 108 37 22];
7344            app.WidthLabel.Text = 'Width';
7345
7346            % Create AxesLabels
7347            app.AxesLabels = uilabel(app.Panel_10);
7348            app.AxesLabels.BackgroundColor = [0.9412 0.9412 0.9412];
7349            app.AxesLabels.FontWeight = 'bold';
7350            app.AxesLabels.Position = [50 297 75 22];
7351            app.AxesLabels.Text = 'Axes Labels';
7352
7353            % Create TextArea2_4
7354            app.TextArea2_4 = uitextarea(app.Panel_10);
7355            app.TextArea2_4.Editable = 'off';
7356            app.TextArea2_4.BackgroundColor = [0.9412 0.9412 0.9412];
7357            app.TextArea2_4.Enable = 'off';
7358            app.TextArea2_4.Position = [173 356 222 109];
7359
7360            % Create xMinLimit
7361            app.xMinLimit = uieditfield(app.Panel_10, 'numeric');
7362            app.xMinLimit.ValueDisplayFormat = '%8.1f';
7363            app.xMinLimit.ValueChangedFcn = createCallbackFcn(app, ↙
@xMinLimitValueChanged, true);
7364            app.xMinLimit.FontColor = [0.651 0.651 0.651];
7365            app.xMinLimit.Enable = 'off';
7366            app.xMinLimit.Tooltip = {'X-axis minimum value'};
7367            app.xMinLimit.Position = [217 356 45 16];
7368
7369            % Create MinimumdepthEditFieldLabel_4
7370            app.MinimumdepthEditFieldLabel_4 = uilabel(app.Panel_10);
7371            app.MinimumdepthEditFieldLabel_4.HorizontalAlignment = 'center';
7372            app.MinimumdepthEditFieldLabel_4.Position = [213 336 51 22];
7373            app.MinimumdepthEditFieldLabel_4.Text = 'X min';
7374
7375            % Create xMaxLimit
7376            app.xMaxLimit = uieditfield(app.Panel_10, 'numeric');
7377            app.xMaxLimit.ValueDisplayFormat = '%8.1f';
7378            app.xMaxLimit.ValueChangedFcn = createCallbackFcn(app, ↙
@xMaxLimitValueChanged, true);
7379            app.xMaxLimit.FontColor = [0.651 0.651 0.651];
7380            app.xMaxLimit.Enable = 'off';
7381            app.xMaxLimit.Tooltip = {'X-axis maximum value'};
7382            app.xMaxLimit.Position = [340 356 55 16];
7383            app.xMaxLimit.Value = 3600;
7384
7385            % Create MaximumdepthEditFieldLabel_5
7386            app.MaximumdepthEditFieldLabel_5 = uilabel(app.Panel_10);
7387            app.MaximumdepthEditFieldLabel_5.HorizontalAlignment = 'center';
7388            app.MaximumdepthEditFieldLabel_5.Position = [343 336 53 22];
7389            app.MaximumdepthEditFieldLabel_5.Text = 'X max';
7390
7391            % Create xLimInterval
7392            app.xLimInterval = uieditfield(app.Panel_10, 'numeric');
7393            app.xLimInterval.Limits = [0 Inf];
7394            app.xLimInterval.ValueDisplayFormat = '%8.1f';
7395            app.xLimInterval.ValueChangedFcn = createCallbackFcn(app, ↙
```

```
@xLimIntervalValueChanged, true);
            app.xLimInterval.FontColor = [0.651 0.651 0.651];
            app.xLimInterval.Enable = 'off';
            app.xLimInterval.Tooltip = {'X-axis tick interval'};
            app.xLimInterval.Position = [275 356 51 16];
            app.xLimInterval.Value = 1000;

            % Create MaximumdepthEditFieldLabel_6
            app.MaximumdepthEditFieldLabel_6 = uilabel(app.Panel_10);
            app.MaximumdepthEditFieldLabel_6.HorizontalAlignment = 'center';
            app.MaximumdepthEditFieldLabel_6.Position = [278 336 45 22];
            app.MaximumdepthEditFieldLabel_6.Text = 'Interval';

            % Create CheckBox_2
            app.CheckBox_2 = uicheckbox(app.Panel_10);
            app.CheckBox_2.ValueChangedFcn = createCallbackFcn(app, ↙
@CheckBox_2ValueChanged, true);
            app.CheckBox_2.Text = '';
            app.CheckBox_2.WordWrap = 'on';
            app.CheckBox_2.FontSize = 10;
            app.CheckBox_2.Position = [173 353 21 22];
            app.CheckBox_2.Value = true;

            % Create MaximumdepthEditFieldLabel_4
            app.MaximumdepthEditFieldLabel_4 = uilabel(app.Panel_10);
            app.MaximumdepthEditFieldLabel_4.HorizontalAlignment = 'right';
            app.MaximumdepthEditFieldLabel_4.Position = [110 446 53 22];
            app.MaximumdepthEditFieldLabel_4.Text = 'Y max';

            % Create yLimInterval
            app.yLimInterval = uieditfield(app.Panel_10, 'numeric');
            app.yLimInterval.Limits = [0 Inf];
            app.yLimInterval.ValueDisplayFormat = '%8.1f';
            app.yLimInterval.ValueChangedFcn = createCallbackFcn(app, ↙
@yLimIntervalValueChanged, true);
            app.yLimInterval.FontColor = [0.651 0.651 0.651];
            app.yLimInterval.Enable = 'off';
            app.yLimInterval.Tooltip = {'Y-axis tick interval'};
            app.yLimInterval.Position = [173 428 36 16];
            app.yLimInterval.Value = 0.5;

            % Create MaximumdepthEditFieldLabel_8
            app.MaximumdepthEditFieldLabel_8 = uilabel(app.Panel_10);
            app.MaximumdepthEditFieldLabel_8.HorizontalAlignment = 'right';
            app.MaximumdepthEditFieldLabel_8.Position = [110 425 53 22];
            app.MaximumdepthEditFieldLabel_8.Text = 'Interval';

            % Create yMinLimit
            app.yMinLimit = uieditfield(app.Panel_10, 'numeric');
            app.yMinLimit.ValueDisplayFormat = '%8.1f';
            app.yMinLimit.ValueChangedFcn = createCallbackFcn(app, ↙
@yMinLimitValueChanged, true);
            app.yMinLimit.FontColor = [0.651 0.651 0.651];
            app.yMinLimit.Enable = 'off';
            app.yMinLimit.Tooltip = {'Y-axis maximum value'};
            app.yMinLimit.Position = [173 407 45 16];
            app.yMinLimit.Value = -1;

```

```
7450            % Create MinimumdepthEditFieldLabel_3
7451            app.MinimumdepthEditFieldLabel_3 = uilabel(app.Panel_10);
7452            app.MinimumdepthEditFieldLabel_3.HorizontalAlignment = 'right';
7453            app.MinimumdepthEditFieldLabel_3.Position = [112 404 51 22];
7454            app.MinimumdepthEditFieldLabel_3.Text = 'Y min';
7455
7456            % Create yMaxLimit
7457            app.yMaxLimit = uieditfield(app.Panel_10, 'numeric');
7458            app.yMaxLimit.Limits = [0 Inf];
7459            app.yMaxLimit.ValueDisplayFormat = '%8.1f';
7460            app.yMaxLimit.ValueChangedFcn = createCallbackFcn(app,↙
@yMaxLimitValueChanged, true);
7461            app.yMaxLimit.FontColor = [0.651 0.651 0.651];
7462            app.yMaxLimit.Enable = 'off';
7463            app.yMaxLimit.Tooltip = {'Y-axis maximum value'};
7464            app.yMaxLimit.Position = [173 449 45 16];
7465            app.yMaxLimit.Value = 1;
7466
7467            % Create CheckBox
7468            app.CheckBox = uicheckbox(app.Panel_10);
7469            app.CheckBox.ValueChangedFcn = createCallbackFcn(app,↙
@CheckBoxValueChanged, true);
7470            app.CheckBox.Text = '';
7471            app.CheckBox.WordWrap = 'on';
7472            app.CheckBox.FontSize = 8;
7473            app.CheckBox.FontWeight = 'bold';
7474            app.CheckBox.Position = [173 370 16 31];
7475            app.CheckBox.Value = true;
7476
7477            % Create YAutoSetLabel
7478            app.YAutoSetLabel = uilabel(app.Panel_10);
7479            app.YAutoSetLabel.HorizontalAlignment = 'right';
7480            app.YAutoSetLabel.FontSize = 10;
7481            app.YAutoSetLabel.Tooltip = {'Automatically set the Y axis↙
limits'};
7482            app.YAutoSetLabel.Position = [110 374 53 22];
7483            app.YAutoSetLabel.Text = 'Y Auto Set';
7484
7485            % Create XAutoSetLabel
7486            app.XAutoSetLabel = uilabel(app.Panel_10);
7487            app.XAutoSetLabel.HorizontalAlignment = 'right';
7488            app.XAutoSetLabel.FontSize = 10;
7489            app.XAutoSetLabel.Tooltip = {'Automatically set the X axis↙
limits'};
7490            app.XAutoSetLabel.Position = [110 353 53 22];
7491            app.XAutoSetLabel.Text = 'X Auto Set';
7492
7493            % Create LabelEditFieldLabel
7494            app.LabelEditFieldLabel = uilabel(app.Panel_10);
7495            app.LabelEditFieldLabel.Position = [50 252 39 22];
7496            app.LabelEditFieldLabel.Text = 'Y Axis';
7497
7498            % Create YAxisEditField
7499            app.YAxisEditField = uieditfield(app.Panel_10, 'text');
7500            app.YAxisEditField.ValueChangedFcn = createCallbackFcn(app,↙
@YAxisEditFieldValueChanged, true);
7501            app.YAxisEditField.FontColor = [0.651 0.651 0.651];
7502            app.YAxisEditField.Position = [105 255 108 16];
```

```
7503            app.YAxisEditField.Value = '\eta (m)';
7504
7505            % Create XandYLimitsLabel_2
7506            app.XandYLimitsLabel_2 = uilabel(app.Panel_10);
7507            app.XandYLimitsLabel_2.BackgroundColor = [0.9412 0.9412 0.9412];
7508            app.XandYLimitsLabel_2.FontWeight = 'bold';
7509            app.XandYLimitsLabel_2.Position = [241 467 91 22];
7510            app.XandYLimitsLabel_2.Text = 'X and Y Limits:';
7511
7512            % Create YAxisLabelEditField_2Label
7513            app.YAxisLabelEditField_2Label = uilabel(app.Panel_10);
7514            app.YAxisLabelEditField_2Label.Position = [50 276 39 22];
7515            app.YAxisLabelEditField_2Label.Text = 'X Axis';
7516
7517            % Create XAxisEditField
7518            app.XAxisEditField = uieditfield(app.Panel_10, 'text');
7519            app.XAxisEditField.ValueChangedFcn = createCallbackFcn(app, ↙
@XAxisEditFieldValueChanged, true);
7520            app.XAxisEditField.ValueChangingFcn = createCallbackFcn(app, ↙
@XAxisEditFieldValueChanging, true);
7521            app.XAxisEditField.FontColor = [0.651 0.651 0.651];
7522            app.XAxisEditField.Position = [105 279 108 16];
7523            app.XAxisEditField.Value = 'Time (sec)';
7524
7525            % Create BackgroundGridLabel
7526            app.BackgroundGridLabel = uilabel(app.Panel_10);
7527            app.BackgroundGridLabel.BackgroundColor = [0.9412 0.9412 0.9412];
7528            app.BackgroundGridLabel.FontWeight = 'bold';
7529            app.BackgroundGridLabel.Position = [288 179 103 22];
7530            app.BackgroundGridLabel.Text = 'Background Grid';
7531
7532            % Create StyleDropDown_4Label
7533            app.StyleDropDown_4Label = uilabel(app.Panel_10);
7534            app.StyleDropDown_4Label.Tooltip = {'Line style of the grid↙
lines'};
7535            app.StyleDropDown_4Label.Position = [289 155 32 22];
7536            app.StyleDropDown_4Label.Text = 'Style';
7537
7538            % Create GridStyle
7539            app.GridStyle = uidropdown(app.Panel_10);
7540            app.GridStyle.Items = {'None', 'X axis only', 'Y axis only', 'Both↙
- major lines', 'Both - with minor lines'};
7541            app.GridStyle.DropDownOpeningFcn = createCallbackFcn(app, ↙
@GridStyleDropDownOpening, true);
7542            app.GridStyle.ValueChangedFcn = createCallbackFcn(app, ↙
@GridStyleValueChanged, true);
7543            app.GridStyle.Tooltip = {''};
7544            app.GridStyle.Position = [364 157 100 17];
7545            app.GridStyle.Value = 'None';
7546
7547            % Create Lineaty0DropDownLabel
7548            app.Lineaty0DropDownLabel = uilabel(app.Panel_10);
7549            app.Lineaty0DropDownLabel.WordWrap = 'on';
7550            app.Lineaty0DropDownLabel.Tooltip = {'Add a horizontal line at y =↙
0'};
7551            app.Lineaty0DropDownLabel.Position = [289 108 69 22];
7552            app.Lineaty0DropDownLabel.Text = 'Line at y=0';
7553
```

```
7554            % Create Lineaty0DropDown
7555            app.Lineaty0DropDown = uidropdown(app.Panel_10);
7556            app.Lineaty0DropDown.Items = {'None', 'Solid', 'Dashed', 'Dotted',↙
'Dash-dotted'};
7557            app.Lineaty0DropDown.ValueChangedFcn = createCallbackFcn(app,↙
@Lineaty0DropDownValueChanged, true);
7558            app.Lineaty0DropDown.Tooltip = {''};
7559            app.Lineaty0DropDown.Position = [364 110 100 17];
7560            app.Lineaty0DropDown.Value = 'Dotted';
7561
7562            % Create LineYThickness
7563            app.LineYThickness = uieditfield(app.Panel_10, 'numeric');
7564            app.LineYThickness.Limits = [0.1 Inf];
7565            app.LineYThickness.ValueDisplayFormat = '%2.1f';
7566            app.LineYThickness.ValueChangedFcn = createCallbackFcn(app,↙
@LineYThicknessValueChanged, true);
7567            app.LineYThickness.HorizontalAlignment = 'left';
7568            app.LineYThickness.FontColor = [0.651 0.651 0.651];
7569            app.LineYThickness.Tooltip = {''};
7570            app.LineYThickness.Position = [364 87 29 16];
7571            app.LineYThickness.Value = 0.5;
7572
7573            % Create ThicknessLabel_3
7574            app.ThicknessLabel_3 = uilabel(app.Panel_10);
7575            app.ThicknessLabel_3.WordWrap = 'on';
7576            app.ThicknessLabel_3.Position = [289 82 71 27];
7577            app.ThicknessLabel_3.Text = 'Width at y=0';
7578
7579            % Create ThicknessLabel_5
7580            app.ThicknessLabel_5 = uilabel(app.Panel_10);
7581            app.ThicknessLabel_5.Position = [50 227 53 22];
7582            app.ThicknessLabel_5.Text = 'Text Size';
7583
7584            % Create GridLabelSize
7585            app.GridLabelSize = uieditfield(app.Panel_10, 'numeric');
7586            app.GridLabelSize.Limits = [1 Inf];
7587            app.GridLabelSize.ValueDisplayFormat = '%2.0f';
7588            app.GridLabelSize.ValueChangedFcn = createCallbackFcn(app,↙
@GridLabelSizeValueChanged, true);
7589            app.GridLabelSize.FontColor = [0.651 0.651 0.651];
7590            app.GridLabelSize.Position = [106 230 29 16];
7591            app.GridLabelSize.Value = 12;
7592
7593            % Create GaugeLineColor_multiple
7594            app.GaugeLineColor_multiple = uidropdown(app.Panel_10);
7595            app.GaugeLineColor_multiple.Items = {'autumn', 'bone',↙
'colorcube', 'cool', 'copper', 'flag', 'gray', 'hot', 'hsv', 'jet', 'lines',↙
'parula', 'pink', 'prism', 'spring', 'summer', 'turbo', 'winter'};
7596            app.GaugeLineColor_multiple.ValueChangedFcn = createCallbackFcn↙
(app, @GaugeLineColor_multipleValueChanged, true);
7597            app.GaugeLineColor_multiple.Position = [108 87 89 17];
7598            app.GaugeLineColor_multiple.Value = 'lines';
7599
7600            % Create FlipCheckBox_3
7601            app.FlipCheckBox_3 = uicheckbox(app.Panel_10);
7602            app.FlipCheckBox_3.Tooltip = {'Reverse the color sequence'};
7603            app.FlipCheckBox_3.Text = 'Flip';
7604            app.FlipCheckBox_3.FontSize = 11;
```

```
7605            app.FlipCheckBox_3.Position = [207 85 40 22];
7606
7607            % Create Height
7608            app.Height = uieditfield(app.Panel_10, 'numeric');
7609            app.Height.Limits = [1 Inf];
7610            app.Height.ValueDisplayFormat = '%2.0f';
7611            app.Height.ValueChangedFcn = createCallbackFcn(app, ↙
@HeightValueChanged, true);
7612            app.Height.FontColor = [0.651 0.651 0.651];
7613            app.Height.Enable = 'off';
7614            app.Height.Tooltip = {''};
7615            app.Height.Position = [283 27 40 16];
7616            app.Height.Value = 8;
7617
7618            % Create ThicknessLabel_6
7619            app.ThicknessLabel_6 = uilabel(app.Panel_10);
7620            app.ThicknessLabel_6.Tooltip = {'Unit: inches'};
7621            app.ThicknessLabel_6.Position = [239 24 41 22];
7622            app.ThicknessLabel_6.Text = 'Height';
7623
7624            % Create FigureSizeLabel
7625            app.FigureSizeLabel = uilabel(app.Panel_10);
7626            app.FigureSizeLabel.FontWeight = 'bold';
7627            app.FigureSizeLabel.Position = [50 43 69 22];
7628            app.FigureSizeLabel.Text = 'Figure Size';
7629
7630            % Create Width
7631            app.Width = uieditfield(app.Panel_10, 'numeric');
7632            app.Width.Limits = [1 Inf];
7633            app.Width.ValueDisplayFormat = '%2.0f';
7634            app.Width.ValueChangedFcn = createCallbackFcn(app, ↙
@WidthValueChanged, true);
7635            app.Width.FontColor = [0.651 0.651 0.651];
7636            app.Width.Enable = 'off';
7637            app.Width.Tooltip = {''};
7638            app.Width.Position = [182 27 41 16];
7639            app.Width.Value = 11;
7640
7641            % Create ThicknessLabel_7
7642            app.ThicknessLabel_7 = uilabel(app.Panel_10);
7643            app.ThicknessLabel_7.Tooltip = {'Unit: inches'};
7644            app.ThicknessLabel_7.Position = [139 24 37 22];
7645            app.ThicknessLabel_7.Text = 'Width';
7646
7647            % Create AutoSetCheckBox
7648            app.AutoSetCheckBox = uicheckbox(app.Panel_10);
7649            app.AutoSetCheckBox.ValueChangedFcn = createCallbackFcn(app, ↙
@AutoSetCheckBoxValueChanged, true);
7650            app.AutoSetCheckBox.Text = 'Auto Set';
7651            app.AutoSetCheckBox.Position = [50 24 68 22];
7652            app.AutoSetCheckBox.Value = true;
7653
7654            % Create CloseFiguresButton_2
7655            app.CloseFiguresButton_2 = uibutton(app.Panel_10, 'push');
7656            app.CloseFiguresButton_2.ButtonPushedFcn = createCallbackFcn(app, ↙
@CloseFiguresButton_2Pushed2, true);
7657            app.CloseFiguresButton_2.Tooltip = {''};
7658            app.CloseFiguresButton_2.Position = [364 24 89 20];
```

```
7659            app.CloseFiguresButton_2.Text = 'Close Figures';
7660
7661            % Create LineYThickness_2
7662            app.LineYThickness_2 = uieditfield(app.Panel_10, 'numeric');
7663            app.LineYThickness_2.Limits = [0.1 Inf];
7664            app.LineYThickness_2.ValueDisplayFormat = '%2.1f';
7665            app.LineYThickness_2.ValueChangedFcn = createCallbackFcn(app,
@LineYThickness_2ValueChanged, true);
7666            app.LineYThickness_2.Editable = 'off';
7667            app.LineYThickness_2.FontColor = [0.651 0.651 0.651];
7668            app.LineYThickness_2.Position = [364 134 29 16];
7669            app.LineYThickness_2.Value = 0.5;
7670
7671            % Create ThicknessLabel_12
7672            app.ThicknessLabel_12 = uilabel(app.Panel_10);
7673            app.ThicknessLabel_12.Position = [289 131 37 22];
7674            app.ThicknessLabel_12.Text = 'Width';
7675
7676            % Create LegendLabel
7677            app.LegendLabel = uilabel(app.Panel_10);
7678            app.LegendLabel.FontWeight = 'bold';
7679            app.LegendLabel.Position = [287 297 48 22];
7680            app.LegendLabel.Text = 'Legend';
7681
7682            % Create legendfirsttextlabel
7683            app.legendfirsttextlabel = uilabel(app.Panel_10);
7684            app.legendfirsttextlabel.Tooltip = {'Text added to data labels,
for example: Station 1, Station 2'};
7685            app.legendfirsttextlabel.Position = [288 276 35 22];
7686
7687            % Create legendfirsttext
7688            app.legendfirsttext = uieditfield(app.Panel_10, 'text');
7689            app.legendfirsttext.ValueChangedFcn = createCallbackFcn(app,
@legendfirsttextValueChanged, true);
7690            app.legendfirsttext.FontColor = [0.651 0.651 0.651];
7691            app.legendfirsttext.Tooltip = {''};
7692            app.legendfirsttext.Position = [345 279 104 16];
7693            app.legendfirsttext.Value = 'Station';
7694
7695            % Create LocationDropDownLabel
7696            app.LocationDropDownLabel = uilabel(app.Panel_10);
7697            app.LocationDropDownLabel.Position = [288 227 52 22];
7698            app.LocationDropDownLabel.Text = 'Location';
7699
7700            % Create LocationDropDown
7701            app.LocationDropDown = uidropdown(app.Panel_10);
7702            app.LocationDropDown.Items = {'North', 'South', 'East', 'West',
'Northeast', 'Northwest', 'Southeast', 'Southwest', 'Northoutside', 'Southoutside',
'Eastoutside', 'Westoutside', 'Northeastoutside', 'Northwestoutside',
'Southeastoutside', 'Southwestoutside', 'Best', 'Bestoutside', 'None'};
7703            app.LocationDropDown.ValueChangedFcn = createCallbackFcn(app,
@LocationDropDownValueChanged, true);
7704            app.LocationDropDown.Tooltip = {''};
7705            app.LocationDropDown.Position = [345 230 104 17];
7706            app.LocationDropDown.Value = 'Best';
7707
7708            % Create LegendSize
7709            app.LegendSize = uieditfield(app.Panel_10, 'numeric');
```

```
7710            app.LegendSize.Limits = [1 Inf];
7711            app.LegendSize.ValueDisplayFormat = '%2.0f';
7712            app.LegendSize.ValueChangedFcn = createCallbackFcn(app, ↙
@LegendSizeValueChanged, true);
7713            app.LegendSize.FontColor = [0.651 0.651 0.651];
7714            app.LegendSize.Position = [345 255 29 16];
7715            app.LegendSize.Value = 12;
7716
7717            % Create ThicknessLabel_4
7718            app.ThicknessLabel_4 = uilabel(app.Panel_10);
7719            app.ThicknessLabel_4.Position = [288 252 53 22];
7720            app.ThicknessLabel_4.Text = 'Text Size';
7721
7722            % Create MarkerDropDownLabel
7723            app.MarkerDropDownLabel = uilabel(app.Panel_10);
7724            app.MarkerDropDownLabel.Position = [50 155 43 22];
7725            app.MarkerDropDownLabel.Text = 'Marker';
7726
7727            % Create MarkerDropDown
7728            app.MarkerDropDown = uidropdown(app.Panel_10);
7729            app.MarkerDropDown.Items = {'none', 'o', '+', '*', '.', 'x', '-',↙
'|', '^', 'v', '>', '<', 'diamond', 'hexagram', 'pentagram', 'square'};
7730            app.MarkerDropDown.Tooltip = {'none'; 'o'; '+'; '*'};
7731            app.MarkerDropDown.Position = [106 157 92 17];
7732            app.MarkerDropDown.Value = 'none';
7733
7734            % Create PLOTButton
7735            app.PLOTButton = uibutton(app.GaugeRecordsTab, 'push');
7736            app.PLOTButton.ButtonPushedFcn = createCallbackFcn(app, ↙
@PLOTButtonPushed, true);
7737            app.PLOTButton.FontSize = 14;
7738            app.PLOTButton.FontWeight = 'bold';
7739            app.PLOTButton.Position = [179 6 183 25];
7740            app.PLOTButton.Text = 'PLOT';
7741
7742            % Create Panel_11
7743            app.Panel_11 = uipanel(app.GaugeRecordsTab);
7744            app.Panel_11.AutoResizeChildren = 'off';
7745            app.Panel_11.SizeChangedFcn = createCallbackFcn(app, ↙
@Panel_11SizeChanged, true);
7746            app.Panel_11.Position = [11 703 520 119];
7747
7748            % Create FilesLabel_2
7749            app.FilesLabel_2 = uilabel(app.Panel_11);
7750            app.FilesLabel_2.FontWeight = 'bold';
7751            app.FilesLabel_2.Tooltip = {'Load the files.'; ''; 'Files must↙
start with ''sta_'''};
7752            app.FilesLabel_2.Position = [44 80 32 22];
7753            app.FilesLabel_2.Text = 'Files';
7754
7755            % Create FileTextArea_2
7756            app.FileTextArea_2 = uitextarea(app.Panel_11);
7757            app.FileTextArea_2.ValueChangedFcn = createCallbackFcn(app, ↙
@FileTextArea_2ValueChanged, true);
7758            app.FileTextArea_2.Editable = 'off';
7759            app.FileTextArea_2.FontColor = [0.651 0.651 0.651];
7760            app.FileTextArea_2.Tooltip = {''};
7761            app.FileTextArea_2.Placeholder = 'sta xxxx';
```

```
7762            app.FileTextArea_2.Position = [44 19 95 60];
7763
7764            % Create Button_16
7765            app.Button_16 = uibutton(app.Panel_11, 'push');
7766            app.Button_16.ButtonPushedFcn = createCallbackFcn(app,
@Button_16Pushed, true);
7767            app.Button_16.FontAngle = 'italic';
7768            app.Button_16.FontColor = [0.651 0.651 0.651];
7769            app.Button_16.Tooltip = {'Load the files.'; ''; 'Files must start
with ''sta_'''};
7770            app.Button_16.Position = [146 60 19 19];
7771            app.Button_16.Text = '...';
7772
7773            % Create XaxisLabel
7774            app.XaxisLabel = uilabel(app.Panel_11);
7775            app.XaxisLabel.Tooltip = {'Select column to use for plotting.';
''; 'Unit for each column: '; '1: time (s)'; '2: eta (m)'; '3: u vector (m/s)'; '4:
v vector (m/s)'; '5: z vector (m/s)'};
7776            app.XaxisLabel.Position = [188 57 44 22];
7777            app.XaxisLabel.Text = 'X axis: ';
7778
7779            % Create SelectColumnLabel
7780            app.SelectColumnLabel = uilabel(app.Panel_11);
7781            app.SelectColumnLabel.FontWeight = 'bold';
7782            app.SelectColumnLabel.Tooltip = {'Select column to use for
plotting.'; ''; 'Unit for each column: '; '1: time (s)'; '2: eta (m)'; '3: u vector
(m/s)'; '4: v vector (m/s)'; '5: z vector (m/s)'};
7783            app.SelectColumnLabel.Position = [188 80 89 22];
7784            app.SelectColumnLabel.Text = 'Select Column';
7785
7786            % Create XaxisdataLabel
7787            app.XaxisdataLabel = uilabel(app.Panel_11);
7788            app.XaxisdataLabel.Tooltip = {'Example:'; 'Convert column 1 data
from seconds to minutes. '};
7789            app.XaxisdataLabel.Position = [357 56 65 22];
7790            app.XaxisdataLabel.Text = 'X axis data';
7791
7792            % Create ConvertLabel
7793            app.ConvertLabel = uilabel(app.Panel_11);
7794            app.ConvertLabel.FontWeight = 'bold';
7795            app.ConvertLabel.Tooltip = {'Examples:'; 'Convert column 1 data
from seconds to minutes. '; ''; 'Convert column 2 data from meters to
centimeters.'};
7796            app.ConvertLabel.Position = [333 80 54 22];
7797            app.ConvertLabel.Text = 'Convert:';
7798
7799            % Create numeratorY
7800            app.numeratorY = uieditfield(app.Panel_11, 'numeric');
7801            app.numeratorY.Limits = [1 Inf];
7802            app.numeratorY.ValueDisplayFormat = '%8.0f';
7803            app.numeratorY.HorizontalAlignment = 'center';
7804            app.numeratorY.FontColor = [0.651 0.651 0.651];
7805            app.numeratorY.Enable = 'off';
7806            app.numeratorY.Position = [445 26 36 16];
7807            app.numeratorY.Value = 100;
7808
7809            % Create denomeratorY
7810            app.denomeratorY = uieditfield(app.Panel_11, 'numeric');
```

```
7811             app.denomeratorY.Limits = [1 Inf];
7812             app.denomeratorY.ValueDisplayFormat = '%8.0f';
7813             app.denomeratorY.ValueChangedFcn = createCallbackFcn(app, ↙
@denomeratorYValueChanged, true);
7814             app.denomeratorY.HorizontalAlignment = 'center';
7815             app.denomeratorY.FontColor = [0.651 0.651 0.651];
7816             app.denomeratorY.Enable = 'off';
7817             app.denomeratorY.Position = [445 11 36 16];
7818             app.denomeratorY.Value = 1;
7819 
7820             % Create Label_9
7821             app.Label_9 = uilabel(app.Panel_11);
7822             app.Label_9.Position = [424 19 13 22];
7823             app.Label_9.Text = '×';
7824 
7825             % Create YaxisEditFieldLabel
7826             app.YaxisEditFieldLabel = uilabel(app.Panel_11);
7827             app.YaxisEditFieldLabel.Tooltip = {'Example: Convert column 2 data↙
from meters to centimeters.'};
7828             app.YaxisEditFieldLabel.Position = [357 19 65 22];
7829             app.YaxisEditFieldLabel.Text = 'Y axis data';
7830 
7831             % Create CheckBox_3
7832             app.CheckBox_3 = uicheckbox(app.Panel_11);
7833             app.CheckBox_3.ValueChangedFcn = createCallbackFcn(app, ↙
@CheckBox_3ValueChanged, true);
7834             app.CheckBox_3.Text = '';
7835             app.CheckBox_3.Position = [333 56 25 22];
7836 
7837             % Create CheckBox_4
7838             app.CheckBox_4 = uicheckbox(app.Panel_11);
7839             app.CheckBox_4.ValueChangedFcn = createCallbackFcn(app, ↙
@CheckBox_4ValueChanged, true);
7840             app.CheckBox_4.Tooltip = {'Only the input in the first textbox↙
under ''Y axis: Use data from column'' will be converted'};
7841             app.CheckBox_4.Text = '';
7842             app.CheckBox_4.Position = [333 19 25 22];
7843 
7844             % Create numeratorX
7845             app.numeratorX = uieditfield(app.Panel_11, 'numeric');
7846             app.numeratorX.Limits = [1 Inf];
7847             app.numeratorX.ValueDisplayFormat = '%8.0f';
7848             app.numeratorX.HorizontalAlignment = 'center';
7849             app.numeratorX.FontColor = [0.651 0.651 0.651];
7850             app.numeratorX.Enable = 'off';
7851             app.numeratorX.Position = [445 64 36 16];
7852             app.numeratorX.Value = 1;
7853 
7854             % Create denomeratorX
7855             app.denomeratorX = uieditfield(app.Panel_11, 'numeric');
7856             app.denomeratorX.Limits = [1 Inf];
7857             app.denomeratorX.ValueDisplayFormat = '%8.0f';
7858             app.denomeratorX.HorizontalAlignment = 'center';
7859             app.denomeratorX.FontColor = [0.651 0.651 0.651];
7860             app.denomeratorX.Enable = 'off';
7861             app.denomeratorX.Position = [445 49 36 16];
7862             app.denomeratorX.Value = 60;
7863 
```

```
7864            % Create Label_5
7865            app.Label_5 = uilabel(app.Panel_11);
7866            app.Label_5.Position = [424 56 13 22];
7867            app.Label_5.Text = '×';
7868
7869            % Create Label_6
7870            app.Label_6 = uilabel(app.Panel_11);
7871            app.Label_6.FontSize = 22;
7872            app.Label_6.Position = [435 51 10 29];
7873            app.Label_6.Text = '(';
7874
7875            % Create Label_7
7876            app.Label_7 = uilabel(app.Panel_11);
7877            app.Label_7.FontSize = 22;
7878            app.Label_7.Position = [487 52 10 29];
7879            app.Label_7.Text = ')';
7880
7881            % Create Label_10
7882            app.Label_10 = uilabel(app.Panel_11);
7883            app.Label_10.FontSize = 22;
7884            app.Label_10.Position = [435 12 10 29];
7885            app.Label_10.Text = '(';
7886
7887            % Create Label_11
7888            app.Label_11 = uilabel(app.Panel_11);
7889            app.Label_11.FontSize = 22;
7890            app.Label_11.Position = [487 13 10 29];
7891            app.Label_11.Text = ')';
7892
7893            % Create YaxisSetcolumncountLabel
7894            app.YaxisSetcolumncountLabel = uilabel(app.Panel_11);
7895            app.YaxisSetcolumncountLabel.Tooltip = {'Select column to use for↙
plotting.'; ''; 'Unit for each column: '; '1: time (s)'; '2: eta (m)'; '3: u vector↙
(m/s)'; '4: v vector (m/s)'; '5: z vector (m/s)'};
7896            app.YaxisSetcolumncountLabel.Position = [188 33 45 22];
7897            app.YaxisSetcolumncountLabel.Text = 'Y axis: ';
7898
7899            % Create YaxisUseDataDropDown
7900            app.YaxisUseDataDropDown = uidropdown(app.Panel_11);
7901            app.YaxisUseDataDropDown.Items = {'1', '2', '3', '4', '5',↙
'3,4,5'};
7902            app.YaxisUseDataDropDown.DropDownOpeningFcn = createCallbackFcn↙
(app, @YaxisUseDataDropDownOpening, true);
7903            app.YaxisUseDataDropDown.ValueChangedFcn = createCallbackFcn(app,↙
@YaxisUseDataDropDownValueChanged, true);
7904            app.YaxisUseDataDropDown.Tooltip = {''};
7905            app.YaxisUseDataDropDown.Position = [231 36 61 16];
7906            app.YaxisUseDataDropDown.Value = '2';
7907
7908            % Create XaxisUseDataDropDown
7909            app.XaxisUseDataDropDown = uidropdown(app.Panel_11);
7910            app.XaxisUseDataDropDown.Items = {'1', '2', '3', '4'};
7911            app.XaxisUseDataDropDown.ValueChangedFcn = createCallbackFcn(app,↙
@XaxisUseDataDropDownValueChanged, true);
7912            app.XaxisUseDataDropDown.Tooltip = {''};
7913            app.XaxisUseDataDropDown.Position = [231 61 41 16];
7914            app.XaxisUseDataDropDown.Value = '1';
7915
```

```
7916            % Create InputDataLabel_3
7917            app.InputDataLabel_3 = uilabel(app.GaugeRecordsTab);
7918            app.InputDataLabel_3.BackgroundColor = [0.9412 0.9412 0.9412];
7919            app.InputDataLabel_3.FontSize = 15;
7920            app.InputDataLabel_3.FontWeight = 'bold';
7921            app.InputDataLabel_3.FontColor = [0.0314 0.3686 0.6];
7922            app.InputDataLabel_3.Position = [38 813 97 22];
7923            app.InputDataLabel_3.Text = '  Input Data  ';
7924
7925            % Create Panel_13
7926            app.Panel_13 = uipanel(app.GaugeRecordsTab);
7927            app.Panel_13.AutoResizeChildren = 'off';
7928            app.Panel_13.Position = [11 39 520 77];
7929
7930            % Create pngCheckBox_2
7931            app.pngCheckBox_2 = uicheckbox(app.Panel_13);
7932            app.pngCheckBox_2.Tooltip = {'Resolution is set to 300 DPI'};
7933            app.pngCheckBox_2.Text = '.png';
7934            app.pngCheckBox_2.Position = [192 12 46 22];
7935
7936            % Create FileFormatLabel_2
7937            app.FileFormatLabel_2 = uilabel(app.Panel_13);
7938            app.FileFormatLabel_2.Position = [68 12 66 22];
7939            app.FileFormatLabel_2.Text = 'File Format';
7940
7941            % Create jpgCheckBox_2
7942            app.jpgCheckBox_2 = uicheckbox(app.Panel_13);
7943            app.jpgCheckBox_2.Tooltip = {'Resolution is set to 300 DPI'};
7944            app.jpgCheckBox_2.Text = '.jpg';
7945            app.jpgCheckBox_2.Position = [139 12 42 22];
7946
7947            % Create pdfCheckBox_2
7948            app.pdfCheckBox_2 = uicheckbox(app.Panel_13);
7949            app.pdfCheckBox_2.Tooltip = {'Vector-type pdf file'};
7950            app.pdfCheckBox_2.Text = '.pdf';
7951            app.pdfCheckBox_2.Position = [249 12 43 22];
7952
7953            % Create Button_18
7954            app.Button_18 = uibutton(app.Panel_13, 'push');
7955            app.Button_18.ButtonPushedFcn = createCallbackFcn(app, ↙
@Button_18Pushed, true);
7956            app.Button_18.FontSize = 10;
7957            app.Button_18.FontAngle = 'italic';
7958            app.Button_18.Tooltip = {'Load the preferred directory.'; '';↙
'"OUTPUT_FILES/Figures"  folder will be created in the selected directory.'};
7959            app.Button_18.Position = [403 42 19 19];
7960            app.Button_18.Text = '...';
7961
7962            % Create OutputDirectoryLabel_3
7963            app.OutputDirectoryLabel_3 = uilabel(app.Panel_13);
7964            app.OutputDirectoryLabel_3.Position = [40 40 94 22];
7965            app.OutputDirectoryLabel_3.Text = 'Output Directory';
7966
7967            % Create OutputDirectoryEditField_2
7968            app.OutputDirectoryEditField_2 = uieditfield(app.Panel_13, ↙
'text');
7969            app.OutputDirectoryEditField_2.ValueChangedFcn = createCallbackFcn↙
(app, @OutputDirectoryEditField_2ValueChanged, true);
```

```
7970            app.OutputDirectoryEditField_2.Editable = 'off';
7971            app.OutputDirectoryEditField_2.FontColor = [0 0 1];
7972            app.OutputDirectoryEditField_2.Tooltip = {'Load the preferred↙
directory.'; ''; '''OUTPUT_FILES/Figures'' folder will be created in the selected↙
directory.'};
7973            app.OutputDirectoryEditField_2.Placeholder = 'Default: Desktop';
7974            app.OutputDirectoryEditField_2.Position = [139 43 257 16];
7975
7976            % Create txtCheckBox
7977            app.txtCheckBox = uicheckbox(app.Panel_13);
7978            app.txtCheckBox.ValueChangedFcn = createCallbackFcn(app,↙
@txtCheckBoxValueChanged, true);
7979            app.txtCheckBox.Enable = 'off';
7980            app.txtCheckBox.Tooltip = {'Save raw data from ''sta_'' files as↙
tab-delimited texts '; ''; 'Functionality enabled when "Plot separately" is↙
selected.'};
7981            app.txtCheckBox.Text = '.txt';
7982            app.txtCheckBox.Position = [356 12 39 22];
7983
7984            % Create epsCheckBox
7985            app.epsCheckBox = uicheckbox(app.Panel_13);
7986            app.epsCheckBox.Tooltip = {'Vector graphic file'};
7987            app.epsCheckBox.Text = '.eps';
7988            app.epsCheckBox.Position = [303 12 45 22];
7989
7990            % Create SavePlotLabel
7991            app.SavePlotLabel = uilabel(app.GaugeRecordsTab);
7992            app.SavePlotLabel.BackgroundColor = [0.9412 0.9412 0.9412];
7993            app.SavePlotLabel.FontSize = 15;
7994            app.SavePlotLabel.FontWeight = 'bold';
7995            app.SavePlotLabel.FontColor = [0.0314 0.3686 0.6];
7996            app.SavePlotLabel.Position = [38 105 92 22];
7997            app.SavePlotLabel.Text = '  Save Plot  ';
7998
7999            % Create PlotStyleandLayoutLabel
8000            app.PlotStyleandLayoutLabel = uilabel(app.GaugeRecordsTab);
8001            app.PlotStyleandLayoutLabel.BackgroundColor = [0.9412 0.9412↙
0.9412];
8002            app.PlotStyleandLayoutLabel.FontSize = 15;
8003            app.PlotStyleandLayoutLabel.FontWeight = 'bold';
8004            app.PlotStyleandLayoutLabel.FontColor = [0.0314 0.3686 0.6];
8005            app.PlotStyleandLayoutLabel.Position = [34 681 174 22];
8006            app.PlotStyleandLayoutLabel.Text = '  Plot Style and Layout  ';
8007
8008            % Create VelocityMapTab
8009            app.VelocityMapTab = uitab(app.TabGroup);
8010            app.VelocityMapTab.AutoResizeChildren = 'off';
8011            app.VelocityMapTab.SizeChangedFcn = createCallbackFcn(app,↙
@VelocityMapTabSizeChanged, true);
8012            app.VelocityMapTab.Title = 'Velocity Map';
8013
8014            % Create GENERATEButton_2
8015            app.GENERATEButton_2 = uibutton(app.VelocityMapTab, 'push');
8016            app.GENERATEButton_2.ButtonPushedFcn = createCallbackFcn(app,↙
@GENERATEButton_2Pushed, true);
8017            app.GENERATEButton_2.FontSize = 14;
8018            app.GENERATEButton_2.FontWeight = 'bold';
8019            app.GENERATEButton_2.Position = [179 6 183 25];
```

```
8020            app.GENERATEButton_2.Text = 'GENERATE';
8021
8022            % Create Panel_14
8023            app.Panel_14 = uipanel(app.VelocityMapTab);
8024            app.Panel_14.AutoResizeChildren = 'off';
8025            app.Panel_14.Position = [11 550 520 65];
8026
8027            % Create QuiverColorDropDown
8028            app.QuiverColorDropDown = uidropdown(app.Panel_14);
8029            app.QuiverColorDropDown.Items = {'Black', 'Dark gray', 'Medium↙
gray', 'Light gray', 'Red', 'Green', 'Blue', 'Yellow', 'Cyan', 'Magenta', 'White'};
8030            app.QuiverColorDropDown.ValueChangedFcn = createCallbackFcn(app,↙
@QuiverColorDropDownValueChanged, true);
8031            app.QuiverColorDropDown.Position = [396 12 87 16];
8032            app.QuiverColorDropDown.Value = 'Black';
8033
8034            % Create XEditField_2Label_3
8035            app.XEditField_2Label_3 = uilabel(app.Panel_14);
8036            app.XEditField_2Label_3.Tooltip = {'Size scaling'};
8037            app.XEditField_2Label_3.Position = [131 9 67 22];
8038            app.XEditField_2Label_3.Text = 'Body Scale';
8039
8040            % Create arrowscale
8041            app.arrowscale = uieditfield(app.Panel_14, 'numeric');
8042            app.arrowscale.Limits = [0 Inf];
8043            app.arrowscale.ValueDisplayFormat = '%8.1f';
8044            app.arrowscale.ValueChangedFcn = createCallbackFcn(app,↙
@arrowscaleValueChanged, true);
8045            app.arrowscale.FontColor = [0.651 0.651 0.651];
8046            app.arrowscale.Tooltip = {'Scaling factor. '; ''; 'Set to 0 to↙
disable auto scaling. '};
8047            app.arrowscale.Position = [228 12 35 16];
8048            app.arrowscale.Value = 10;
8049
8050            % Create ArrowThicknessLabel
8051            app.ArrowThicknessLabel = uilabel(app.Panel_14);
8052            app.ArrowThicknessLabel.HorizontalAlignment = 'right';
8053            app.ArrowThicknessLabel.Tooltip = {'Line thickness'};
8054            app.ArrowThicknessLabel.Position = [281 9 37 22];
8055            app.ArrowThicknessLabel.Text = 'Width';
8056
8057            % Create ArrowThickness
8058            app.ArrowThickness = uieditfield(app.Panel_14, 'numeric');
8059            app.ArrowThickness.Limits = [0 Inf];
8060            app.ArrowThickness.ValueDisplayFormat = '%.1f';
8061            app.ArrowThickness.ValueChangedFcn = createCallbackFcn(app,↙
@ArrowThicknessValueChanged, true);
8062            app.ArrowThickness.FontColor = [0.651 0.651 0.651];
8063            app.ArrowThickness.Tooltip = {''};
8064            app.ArrowThickness.Position = [335 12 35 16];
8065            app.ArrowThickness.Value = 1;
8066
8067            % Create ColorLabel_2
8068            app.ColorLabel_2 = uilabel(app.Panel_14);
8069            app.ColorLabel_2.Position = [397 30 35 22];
8070            app.ColorLabel_2.Text = 'Color';
8071
8072            % Create ArrowSizeLabel
```

```
8073            app.ArrowSizeLabel = uilabel(app.Panel_14);
8074            app.ArrowSizeLabel.WordWrap = 'on';
8075            app.ArrowSizeLabel.Tooltip = {'Arrowhead size relative to the body size'};
8076            app.ArrowSizeLabel.Position = [131 27 85 29];
8077            app.ArrowSizeLabel.Text = 'Arrowhead Size';
8078
8079            % Create ArrowHeadSize
8080            app.ArrowHeadSize = uieditfield(app.Panel_14, 'numeric');
8081            app.ArrowHeadSize.Limits = [0 1];
8082            app.ArrowHeadSize.ValueDisplayFormat = '%.1f';
8083            app.ArrowHeadSize.ValueChangedFcn = createCallbackFcn(app, @ArrowHeadSizeValueChanged, true);
8084            app.ArrowHeadSize.FontColor = [0.651 0.651 0.651];
8085            app.ArrowHeadSize.Tooltip = {''};
8086            app.ArrowHeadSize.Position = [228 33 35 16];
8087            app.ArrowHeadSize.Value = 0.2;
8088
8089            % Create PlotvectorsCheckBox
8090            app.PlotvectorsCheckBox = uicheckbox(app.Panel_14);
8091            app.PlotvectorsCheckBox.ValueChangedFcn = createCallbackFcn(app, @PlotvectorsCheckBoxValueChanged, true);
8092            app.PlotvectorsCheckBox.Tooltip = {'Overlay vectors on the map. '; ''; 'Applicable when U and V vectors are loaded.'};
8093            app.PlotvectorsCheckBox.Text = 'Plot vectors';
8094            app.PlotvectorsCheckBox.FontWeight = 'bold';
8095            app.PlotvectorsCheckBox.Position = [31 19 91 22];
8096
8097            % Create SpacingLabel
8098            app.SpacingLabel = uilabel(app.Panel_14);
8099            app.SpacingLabel.HorizontalAlignment = 'right';
8100            app.SpacingLabel.Tooltip = {'Spacing between arrows to avoid clustering on the map'};
8101            app.SpacingLabel.Position = [278 30 50 22];
8102            app.SpacingLabel.Text = 'Spacing';
8103
8104            % Create ArrowSpacing
8105            app.ArrowSpacing = uieditfield(app.Panel_14, 'numeric');
8106            app.ArrowSpacing.Limits = [0.001 Inf];
8107            app.ArrowSpacing.ValueDisplayFormat = '%11.2f';
8108            app.ArrowSpacing.ValueChangedFcn = createCallbackFcn(app, @ArrowSpacingValueChanged, true);
8109            app.ArrowSpacing.FontColor = [0.651 0.651 0.651];
8110            app.ArrowSpacing.Position = [337 33 35 16];
8111            app.ArrowSpacing.Value = 1;
8112
8113            % Create Panel_15
8114            app.Panel_15 = uipanel(app.VelocityMapTab);
8115            app.Panel_15.AutoResizeChildren = 'off';
8116            app.Panel_15.SizeChangedFcn = createCallbackFcn(app, @Panel_15SizeChanged, true);
8117            app.Panel_15.Position = [11 626 520 196];
8118
8119            % Create FileTextArea_3
8120            app.FileTextArea_3 = uitextarea(app.Panel_15);
8121            app.FileTextArea_3.ValueChangedFcn = createCallbackFcn(app, @FileTextArea_3ValueChanged, true);
8122            app.FileTextArea_3.Editable = 'off';
```

```
8123            app.FileTextArea_3.Tooltip = {''};
8124            app.FileTextArea_3.Placeholder = 'u_xxxx; umean_xxxx; umax_xxxx';
8125            app.FileTextArea_3.Position = [32 77 84 74];
8126
8127            % Create UvectorsLabel
8128            app.UvectorsLabel = uilabel(app.Panel_15);
8129            app.UvectorsLabel.WordWrap = 'on';
8130            app.UvectorsLabel.FontWeight = 'bold';
8131            app.UvectorsLabel.Tooltip = {'Load files with filenames starting
with ''u'''};
8132            app.UvectorsLabel.Position = [33 157 61 22];
8133            app.UvectorsLabel.Text = 'U vectors';
8134
8135            % Create DepthFileEditField_2
8136            app.DepthFileEditField_2 = uieditfield(app.Panel_15, 'text');
8137            app.DepthFileEditField_2.ValueChangedFcn = createCallbackFcn(app,
@DepthFileEditField_2ValueChanged, true);
8138            app.DepthFileEditField_2.Editable = 'off';
8139            app.DepthFileEditField_2.Tooltip = {'Load the bathymetry file used
in the simulation'; ''; 'Format accepted: .txt, .tif'};
8140            app.DepthFileEditField_2.Placeholder = '.txt, .tif., mask_';
8141            app.DepthFileEditField_2.Position = [33 22 96 16];
8142
8143            % Create BathymetryLabel_2
8144            app.BathymetryLabel_2 = uilabel(app.Panel_15);
8145            app.BathymetryLabel_2.FontWeight = 'bold';
8146            app.BathymetryLabel_2.Tooltip = {'Load the file'; ''; 'Select the
bathymetry used in the simulation'; ''; 'Format accepted: .txt, .tif'};
8147            app.BathymetryLabel_2.Position = [33 38 71 22];
8148            app.BathymetryLabel_2.Text = 'Bathymetry';
8149
8150            % Create Button_22
8151            app.Button_22 = uibutton(app.Panel_15, 'push');
8152            app.Button_22.ButtonPushedFcn = createCallbackFcn(app,
@Button_22Pushed, true);
8153            app.Button_22.Tooltip = {'Load the file'; ''; 'Select the
bathymetry used in the simulation'; ''; 'Format accepted: .txt, .tif'};
8154            app.Button_22.Position = [136 20 19 19];
8155            app.Button_22.Text = '...';
8156
8157            % Create SouthwestCornerLabel_2
8158            app.SouthwestCornerLabel_2 = uilabel(app.Panel_15);
8159            app.SouthwestCornerLabel_2.FontWeight = 'bold';
8160            app.SouthwestCornerLabel_2.Tooltip = {'Coordinates of the
southwest corner of the loaded files.'};
8161            app.SouthwestCornerLabel_2.Position = [244 134 110 22];
8162            app.SouthwestCornerLabel_2.Text = 'Southwest Corner';
8163
8164            % Create LongitudeEditField_2Label
8165            app.LongitudeEditField_2Label = uilabel(app.Panel_15);
8166            app.LongitudeEditField_2Label.HorizontalAlignment = 'right';
8167            app.LongitudeEditField_2Label.WordWrap = 'on';
8168            app.LongitudeEditField_2Label.Tooltip = {'West boundary'; '';
'Unit: degrees / meters'};
8169            app.LongitudeEditField_2Label.Position = [246 114 60 22];
8170            app.LongitudeEditField_2Label.Text = 'Longitude';
8171
8172            % Create LongitudeEditField_2
```

```
8173            app.LongitudeEditField_2 = uieditfield(app.Panel_15, 'numeric');
8174            app.LongitudeEditField_2.ValueDisplayFormat = '%8.4f';
8175            app.LongitudeEditField_2.ValueChangedFcn = createCallbackFcn(app,
@LongitudeEditField_2ValueChanged, true);
8176            app.LongitudeEditField_2.FontColor = [0.651 0.651 0.651];
8177            app.LongitudeEditField_2.Tooltip = {''};
8178            app.LongitudeEditField_2.Position = [314 117 60 16];
8179
8180            % Create LatitudeEditField_2Label
8181            app.LatitudeEditField_2Label = uilabel(app.Panel_15);
8182            app.LatitudeEditField_2Label.HorizontalAlignment = 'right';
8183            app.LatitudeEditField_2Label.WordWrap = 'on';
8184            app.LatitudeEditField_2Label.Tooltip = {'South boundary'; '';
'Unit: degrees / meters'};
8185            app.LatitudeEditField_2Label.Position = [251 94 44 17];
8186            app.LatitudeEditField_2Label.Text = 'Latitude';
8187
8188            % Create LatitudeEditField_2
8189            app.LatitudeEditField_2 = uieditfield(app.Panel_15, 'numeric');
8190            app.LatitudeEditField_2.ValueDisplayFormat = '%8.4f';
8191            app.LatitudeEditField_2.ValueChangedFcn = createCallbackFcn(app,
@LatitudeEditField_2ValueChanged, true);
8192            app.LatitudeEditField_2.FontColor = [0.651 0.651 0.651];
8193            app.LatitudeEditField_2.Tooltip = {''};
8194            app.LatitudeEditField_2.Position = [314 95 60 16];
8195
8196            % Create GridSizeLabel
8197            app.GridSizeLabel = uilabel(app.Panel_15);
8198            app.GridSizeLabel.FontWeight = 'bold';
8199            app.GridSizeLabel.Tooltip = {'Resolution'; 'Unit: degrees /
meters'};
8200            app.GridSizeLabel.Position = [401 134 57 22];
8201            app.GridSizeLabel.Text = 'Grid Size';
8202
8203            % Create YEditField_2Label_2
8204            app.YEditField_2Label_2 = uilabel(app.Panel_15);
8205            app.YEditField_2Label_2.HorizontalAlignment = 'right';
8206            app.YEditField_2Label_2.Tooltip = {'Y direction'; ''; 'Unit:
degrees / meters'};
8207            app.YEditField_2Label_2.Position = [403 92 13 21];
8208            app.YEditField_2Label_2.Text = 'Y';
8209
8210            % Create gridY_2
8211            app.gridY_2 = uieditfield(app.Panel_15, 'numeric');
8212            app.gridY_2.Limits = [0 Inf];
8213            app.gridY_2.ValueDisplayFormat = '%8.5f';
8214            app.gridY_2.ValueChangedFcn = createCallbackFcn(app,
@gridY_2ValueChanged, true);
8215            app.gridY_2.FontColor = [0.651 0.651 0.651];
8216            app.gridY_2.Tooltip = {''};
8217            app.gridY_2.Position = [427 94 60 16];
8218
8219            % Create TotalSimuilationTimesecEditFieldLabel_4
8220            app.TotalSimuilationTimesecEditFieldLabel_4 = uilabel(app.
Panel_15);
8221            app.TotalSimuilationTimesecEditFieldLabel_4.FontWeight = 'bold';
8222            app.TotalSimuilationTimesecEditFieldLabel_4.Position = [244 38 98
22];
```

```
8223            app.TotalSimuilationTimesecEditFieldLabel_4.Text = 'Simulation Time';
8224
8225            % Create StartTime2
8226            app.StartTime2 = uieditfield(app.Panel_15, 'numeric');
8227            app.StartTime2.Limits = [0 Inf];
8228            app.StartTime2.ValueDisplayFormat = '%8.1f';
8229            app.StartTime2.ValueChangedFcn = createCallbackFcn(app, @StartTime2ValueChanged, true);
8230            app.StartTime2.FontColor = [0.651 0.651 0.651];
8231            app.StartTime2.Tooltip = {''};
8232            app.StartTime2.Position = [314 22 47 16];
8233
8234            % Create StartLabel_2
8235            app.StartLabel_2 = uilabel(app.Panel_15);
8236            app.StartLabel_2.HorizontalAlignment = 'right';
8237            app.StartLabel_2.Tooltip = {'For imported initial tsunami files with a non-zero start time, input the value here. Example: for a 30-second snapshot, input ''30''.'; ''; 'Unit: seconds'};
8238            app.StartLabel_2.Position = [275 19 31 22];
8239            app.StartLabel_2.Text = 'Start';
8240
8241            % Create IntervalLabel_2
8242            app.IntervalLabel_2 = uilabel(app.Panel_15);
8243            app.IntervalLabel_2.HorizontalAlignment = 'right';
8244            app.IntervalLabel_2.Tooltip = {'Snapshot interval.'; ''; 'Same as PLOT_INTV in input.txt'; ''; 'Unit: seconds'};
8245            app.IntervalLabel_2.Position = [372 19 45 22];
8246            app.IntervalLabel_2.Text = 'Interval';
8247
8248            % Create TotalSimuilationTimesecEditField_4
8249            app.TotalSimuilationTimesecEditField_4 = uieditfield(app.Panel_15, 'numeric');
8250            app.TotalSimuilationTimesecEditField_4.Limits = [1 Inf];
8251            app.TotalSimuilationTimesecEditField_4.ValueDisplayFormat = '%8.1f';
8252            app.TotalSimuilationTimesecEditField_4.ValueChangedFcn = createCallbackFcn(app, @TotalSimuilationTimesecEditField_4ValueChanged, true);
8253            app.TotalSimuilationTimesecEditField_4.FontColor = [0.651 0.651 0.651];
8254            app.TotalSimuilationTimesecEditField_4.Tooltip = {''};
8255            app.TotalSimuilationTimesecEditField_4.Position = [427 22 60 16];
8256            app.TotalSimuilationTimesecEditField_4.Value = 1;
8257
8258            % Create XEditField_2Label_2
8259            app.XEditField_2Label_2 = uilabel(app.Panel_15);
8260            app.XEditField_2Label_2.HorizontalAlignment = 'right';
8261            app.XEditField_2Label_2.Tooltip = {'X direction'; ''; 'Unit: degrees / meters'};
8262            app.XEditField_2Label_2.Position = [406 114 13 22];
8263            app.XEditField_2Label_2.Text = 'X ';
8264
8265            % Create gridX_2
8266            app.gridX_2 = uieditfield(app.Panel_15, 'numeric');
8267            app.gridX_2.Limits = [0 Inf];
8268            app.gridX_2.ValueDisplayFormat = '%8.5f';
8269            app.gridX_2.ValueChangedFcn = createCallbackFcn(app, @gridX_2ValueChanged, true);
```

```
8270            app.gridX_2.FontColor = [0.651 0.651 0.651];
8271            app.gridX_2.Tooltip = {''};
8272            app.gridX_2.Position = [427 117 60 16];
8273
8274            % Create FileTextArea_4
8275            app.FileTextArea_4 = uitextarea(app.Panel_15);
8276            app.FileTextArea_4.Editable = 'off';
8277            app.FileTextArea_4.Tooltip = {'The V-vector file is automatically
searched in the same directory as the corresponding U-vector file.'};
8278            app.FileTextArea_4.Placeholder = '(Auto finds) v_xxxx;
vmean_xxxx;';
8279            app.FileTextArea_4.Position = [127 77 84 74];
8280
8281            % Create Button_19
8282            app.Button_19 = uibutton(app.Panel_15, 'push');
8283            app.Button_19.ButtonPushedFcn = createCallbackFcn(app,
@Button_19Pushed, true);
8284            app.Button_19.FontAngle = 'italic';
8285            app.Button_19.FontColor = [0.651 0.651 0.651];
8286            app.Button_19.Tooltip = {'Load files with filenames starting with
''u'''};
8287            app.Button_19.Position = [93 159 19 19];
8288            app.Button_19.Text = '...';
8289
8290            % Create InputDataLabel_4
8291            app.InputDataLabel_4 = uilabel(app.VelocityMapTab);
8292            app.InputDataLabel_4.BackgroundColor = [0.9412 0.9412 0.9412];
8293            app.InputDataLabel_4.FontSize = 15;
8294            app.InputDataLabel_4.FontWeight = 'bold';
8295            app.InputDataLabel_4.FontColor = [0.0314 0.3686 0.6];
8296            app.InputDataLabel_4.Position = [38 813 97 22];
8297            app.InputDataLabel_4.Text = '  Input Data  ';
8298
8299            % Create ArrowsLabel
8300            app.ArrowsLabel = uilabel(app.VelocityMapTab);
8301            app.ArrowsLabel.BackgroundColor = [0.9412 0.9412 0.9412];
8302            app.ArrowsLabel.FontSize = 15;
8303            app.ArrowsLabel.FontWeight = 'bold';
8304            app.ArrowsLabel.FontColor = [0.0314 0.3686 0.6];
8305            app.ArrowsLabel.Position = [38 603 73 22];
8306            app.ArrowsLabel.Text = '  Arrows  ';
8307
8308            % Create Panel_16
8309            app.Panel_16 = uipanel(app.VelocityMapTab);
8310            app.Panel_16.AutoResizeChildren = 'off';
8311            app.Panel_16.SizeChangedFcn = createCallbackFcn(app,
@Panel_16SizeChanged, true);
8312            app.Panel_16.Position = [11 315 520 222];
8313
8314            % Create TabGroup3
8315            app.TabGroup3 = uitabgroup(app.Panel_16);
8316            app.TabGroup3.AutoResizeChildren = 'off';
8317            app.TabGroup3.Position = [0 -1 520 204];
8318
8319            % Create BasemapTab
8320            app.BasemapTab = uitab(app.TabGroup3);
8321            app.BasemapTab.AutoResizeChildren = 'off';
8322            app.BasemapTab.SizeChangedFcn = createCallbackFcn(app,
```

```
@BasemapTabSizeChanged, true);
8323             app.BasemapTab.Title = 'Basemap';
8324
8325             % Create ButtonGroup_16
8326             app.ButtonGroup_16 = uibuttongroup(app.BasemapTab);
8327             app.ButtonGroup_16.AutoResizeChildren = 'off';
8328             app.ButtonGroup_16.SelectionChangedFcn = createCallbackFcn(app, ↙
@ButtonGroup_16SelectionChanged, true);
8329             app.ButtonGroup_16.BorderType = 'none';
8330             app.ButtonGroup_16.FontWeight = 'bold';
8331             app.ButtonGroup_16.Position = [14 54 117 120];
8332
8333             % Create VelocityButton
8334             app.VelocityButton = uiradiobutton(app.ButtonGroup_16);
8335             app.VelocityButton.Tooltip = {'Z component of the velocity'};
8336             app.VelocityButton.Text = 'Z Velocity';
8337             app.VelocityButton.FontWeight = 'bold';
8338             app.VelocityButton.Position = [11 98 78 22];
8339             app.VelocityButton.Value = true;
8340
8341             % Create etaButton
8342             app.etaButton = uiradiobutton(app.ButtonGroup_16);
8343             app.etaButton.Tooltip = {'Sea surface displacement at  specific↙
time snapshot.'; ''; 'The eta files are automatically loaded. '; ''; 'Ensure the↙
eta files are in the same folder as the U vector files.'};
8344             app.etaButton.Text = 'eta';
8345             app.etaButton.FontWeight = 'bold';
8346             app.etaButton.Position = [10 74 40 22];
8347
8348             % Create hmaxButton
8349             app.hmaxButton = uiradiobutton(app.ButtonGroup_16);
8350             app.hmaxButton.Tooltip = {'Maximum wave height.'; ''; ''; 'The↙
hmax files are automatically loaded. '; ''; 'Ensure the hmax files are in the same↙
folder as the U vector files.'};
8351             app.hmaxButton.Text = 'hmax';
8352             app.hmaxButton.FontWeight = 'bold';
8353             app.hmaxButton.Position = [9 50 53 22];
8354
8355             % Create BathymetryButton
8356             app.BathymetryButton = uiradiobutton(app.ButtonGroup_16);
8357             app.BathymetryButton.Text = 'Bathymetry';
8358             app.BathymetryButton.FontWeight = 'bold';
8359             app.BathymetryButton.Position = [8 26 88 22];
8360
8361             % Create VorticityButton
8362             app.VorticityButton = uiradiobutton(app.ButtonGroup_16);
8363             app.VorticityButton.Text = 'Vorticity';
8364             app.VorticityButton.FontWeight = 'bold';
8365             app.VorticityButton.Position = [7 3 69 22];
8366
8367             % Create ThirdTabColorMapPanel
8368             app.ThirdTabColorMapPanel = uipanel(app.BasemapTab);
8369             app.ThirdTabColorMapPanel.AutoResizeChildren = 'off';
8370             app.ThirdTabColorMapPanel.BorderType = 'none';
8371             app.ThirdTabColorMapPanel.Position = [265 32 203 113];
8372
8373             % Create maplabelsize
8374             app.maplabelsize = uieditfield(app.ThirdTabColorMapPanel, ↙
```

```
'numeric');
8375             app.maplabelsize.Limits = [0 50];
8376             app.maplabelsize.ValueDisplayFormat = '%3.0f';
8377             app.maplabelsize.ValueChangedFcn = createCallbackFcn(app,
@maplabelsizeValueChanged, true);
8378             app.maplabelsize.FontColor = [0.651 0.651 0.651];
8379             app.maplabelsize.Tooltip = {''};
8380             app.maplabelsize.Position = [70 11 35 16];
8381             app.maplabelsize.Value = 10;
8382
8383             % Create BackgroundMapColorDropDown
8384             app.BackgroundMapColorDropDown = uidropdown(app.
ThirdTabColorMapPanel);
8385             app.BackgroundMapColorDropDown.Items = {'--- MATLAB default ----',
'autumn', 'bone', 'colorcube', 'cool', 'copper', 'flag', 'gray', 'hot', 'hsv',
'jet', 'parula', 'pink', 'prism', 'spring', 'summer', 'turbo', 'winter', '---
CBREWER 2 ---', '< sequential >', 'blue', 'blue - green', 'blue - purple', 'green -
blue', 'greens', 'grays', 'oranges', 'orange - red', 'purple - blue', 'purple -
blue - green', 'purple - red', 'purples', 'red - purple', 'reds', 'yellow - green',
'yellow - green - blue', 'yellow - orange - brown', 'yellow - orange - red', '<
divergent >', 'brown - teal', 'pink - light green', 'purple - green', 'purple -
orange', 'red - blue', 'red - gray', 'red - yellow - blue', 'red - yellow - green',
'spectral', '< qualitative >', 'accent', 'dark 2', 'paired', 'pastel 1', 'pastel
2', 'set 1', 'set 2', 'set 3'};
8386             app.BackgroundMapColorDropDown.DropDownOpeningFcn =
createCallbackFcn(app, @BackgroundMapColorDropDownOpening, true);
8387             app.BackgroundMapColorDropDown.ValueChangedFcn = createCallbackFcn
(app, @BackgroundMapColorDropDownValueChanged, true);
8388             app.BackgroundMapColorDropDown.Position = [8 92 174 16];
8389             app.BackgroundMapColorDropDown.Value = 'blue - purple';
8390
8391             % Create FlipCheckBox_4
8392             app.FlipCheckBox_4 = uicheckbox(app.ThirdTabColorMapPanel);
8393             app.FlipCheckBox_4.Tooltip = {'Reverse the color sequence'};
8394             app.FlipCheckBox_4.Text = 'Flip';
8395             app.FlipCheckBox_4.FontSize = 11;
8396             app.FlipCheckBox_4.Position = [8 67 40 22];
8397
8398             % Create InterpolationDivisionEditFieldLabel_2
8399             app.InterpolationDivisionEditFieldLabel_2 = uilabel(app.
ThirdTabColorMapPanel);
8400             app.InterpolationDivisionEditFieldLabel_2.HorizontalAlignment =
'right';
8401             app.InterpolationDivisionEditFieldLabel_2.FontSize = 11;
8402             app.InterpolationDivisionEditFieldLabel_2.Tooltip = {'Defines the
color distribution:'; ''; 'Low value:'; 'Distinct color boundaries'; ''; 'Higher
value:'; 'Smoother color transition'};
8403             app.InterpolationDivisionEditFieldLabel_2.Position = [93 68 48
22];
8404             app.InterpolationDivisionEditFieldLabel_2.Text = 'Division';
8405
8406             % Create InterpolationDivisionEditField_2
8407             app.InterpolationDivisionEditField_2 = uieditfield(app.
ThirdTabColorMapPanel, 'numeric');
8408             app.InterpolationDivisionEditField_2.Limits = [0 Inf];
8409             app.InterpolationDivisionEditField_2.RoundFractionalValues = 'on';
8410             app.InterpolationDivisionEditField_2.ValueDisplayFormat = '%3.0f';
8411             app.InterpolationDivisionEditField_2.ValueChangedFcn =
```

```
createCallbackFcn(app, @InterpolationDivisionEditField_2ValueChanged, true);
            app.InterpolationDivisionEditField_2.FontColor = [0.651 0.651 0.651];
            app.InterpolationDivisionEditField_2.Tooltip = {''};
            app.InterpolationDivisionEditField_2.Position = [146 72 36 16];
            app.InterpolationDivisionEditField_2.Value = 4;

            % Create MinBarValue
            app.MinBarValue = uieditfield(app.ThirdTabColorMapPanel, 'numeric');
            app.MinBarValue.ValueDisplayFormat = '%.2f';
            app.MinBarValue.ValueChangedFcn = createCallbackFcn(app, @MinBarValueValueChanged, true);
            app.MinBarValue.FontColor = [0.651 0.651 0.651];
            app.MinBarValue.Tooltip = {''};
            app.MinBarValue.Position = [50 34 55 16];

            % Create toLabel_3
            app.toLabel_3 = uilabel(app.ThirdTabColorMapPanel);
            app.toLabel_3.Tooltip = {'Define the minimum value to display in the map.'; ''; 'Unit: meters'};
            app.toLabel_3.Position = [111 31 25 22];
            app.toLabel_3.Text = 'to';

            % Create mLabel_3
            app.mLabel_3 = uilabel(app.ThirdTabColorMapPanel);
            app.mLabel_3.Tooltip = {'Define the maximum value to display in the map.'; ''; 'Unit: meters'};
            app.mLabel_3.Position = [186 31 25 22];
            app.mLabel_3.Text = 'm';

            % Create MaxBarValue
            app.MaxBarValue = uieditfield(app.ThirdTabColorMapPanel, 'numeric');
            app.MaxBarValue.ValueDisplayFormat = '%.2f';
            app.MaxBarValue.ValueChangedFcn = createCallbackFcn(app, @MaxBarValueValueChanged, true);
            app.MaxBarValue.FontColor = [0.651 0.651 0.651];
            app.MaxBarValue.Tooltip = {''};
            app.MaxBarValue.Position = [127 34 55 16];
            app.MaxBarValue.Value = 6;

            % Create LimitsLabel
            app.LimitsLabel = uilabel(app.ThirdTabColorMapPanel);
            app.LimitsLabel.Tooltip = {'Define the maximum value to display in the map.'; ''; 'Unit: meters'};
            app.LimitsLabel.Position = [8 31 38 22];
            app.LimitsLabel.Text = 'Limits';

            % Create TextSizeLabel
            app.TextSizeLabel = uilabel(app.ThirdTabColorMapPanel);
            app.TextSizeLabel.Tooltip = {'Define the maximum value to display in the map.'; ''; 'Unit: meters'};
            app.TextSizeLabel.Position = [8 8 53 22];
            app.TextSizeLabel.Text = 'Text Size';

            % Create ColormapLabel
            app.ColormapLabel = uilabel(app.BasemapTab);
```

```
8460            app.ColormapLabel.FontWeight = 'bold';
8461            app.ColormapLabel.Position = [195 120 62 22];
8462            app.ColormapLabel.Text = 'Colormap';
8463
8464            % Create ColorbarLabel
8465            app.ColorbarLabel = uilabel(app.BasemapTab);
8466            app.ColorbarLabel.FontWeight = 'bold';
8467            app.ColorbarLabel.Position = [195 62 56 22];
8468            app.ColorbarLabel.Text = 'Colorbar';
8469
8470            % Create FileTextArea_5
8471            app.FileTextArea_5 = uitextarea(app.BasemapTab);
8472            app.FileTextArea_5.ValueChangedFcn = createCallbackFcn(app,↙
@FileTextArea_5ValueChanged, true);
8473            app.FileTextArea_5.Editable = 'off';
8474            app.FileTextArea_5.FontColor = [0.149 0.149 0.149];
8475            app.FileTextArea_5.BackgroundColor = [0.9412 0.9412 0.9412];
8476            app.FileTextArea_5.Tooltip = {'File preview. '; ''; 'Ensure that↙
the listed files are in the same folder as the U vector files.'};
8477            app.FileTextArea_5.Placeholder = '(Filename preview)';
8478            app.FileTextArea_5.Position = [21 16 114 35];
8479
8480            % Create HorizontalCheckBox_2
8481            app.HorizontalCheckBox_2 = uicheckbox(app.BasemapTab);
8482            app.HorizontalCheckBox_2.Text = 'Horizontal';
8483            app.HorizontalCheckBox_2.Position = [302 148 76 22];
8484
8485            % Create VerticalCheckBox_2
8486            app.VerticalCheckBox_2 = uicheckbox(app.BasemapTab);
8487            app.VerticalCheckBox_2.Text = 'Vertical';
8488            app.VerticalCheckBox_2.Position = [388 148 61 22];
8489
8490            % Create FlipBasemapLabel
8491            app.FlipBasemapLabel = uilabel(app.BasemapTab);
8492            app.FlipBasemapLabel.FontWeight = 'bold';
8493            app.FlipBasemapLabel.Tooltip = {'Flips the basemap.'; '';↙
'Horizontally (left-right)'; 'Vertically (up-side down)'};
8494            app.FlipBasemapLabel.Position = [195 148 83 22];
8495            app.FlipBasemapLabel.Text = 'Flip Basemap';
8496
8497            % Create CoastlinecolourDropDownLabel_12
8498            app.CoastlinecolourDropDownLabel_12 = uilabel(app.BasemapTab);
8499            app.CoastlinecolourDropDownLabel_12.FontWeight = 'bold';
8500            app.CoastlinecolourDropDownLabel_12.Tooltip = {'Required input:';↙
'Bathymetry file'};
8501            app.CoastlinecolourDropDownLabel_12.Position = [195 7 69 22];
8502            app.CoastlinecolourDropDownLabel_12.Text = 'Land Color';
8503
8504            % Create LandColor
8505            app.LandColor = uidropdown(app.BasemapTab);
8506            app.LandColor.Items = {'Black', 'Dark gray', 'Medium gray', 'Light↙
gray', 'Red', 'Green', 'Blue', 'Yellow', 'Cyan', 'Magenta', 'White'};
8507            app.LandColor.Enable = 'off';
8508            app.LandColor.Tooltip = {''};
8509            app.LandColor.Position = [273 10 105 16];
8510            app.LandColor.Value = 'Dark gray';
8511
8512            % Create BathymetryContoursTab_2
```

```
8513            app.BathymetryContoursTab_2 = uitab(app.TabGroup3);
8514            app.BathymetryContoursTab_2.AutoResizeChildren = 'off';
8515            app.BathymetryContoursTab_2.Tooltip = {'Enabled when bathymetry
file is uploaded'};
8516            app.BathymetryContoursTab_2.Title = 'Bathymetry Contours';
8517
8518            % Create RangeLabel
8519            app.RangeLabel = uilabel(app.BathymetryContoursTab_2);
8520            app.RangeLabel.FontWeight = 'bold';
8521            app.RangeLabel.Position = [45 109 42 22];
8522            app.RangeLabel.Text = 'Range';
8523
8524            % Create PlotBathymetryContoursCheckBox
8525            app.PlotBathymetryContoursCheckBox = uicheckbox(app.
BathymetryContoursTab_2);
8526            app.PlotBathymetryContoursCheckBox.ValueChangedFcn =
createCallbackFcn(app, @PlotBathymetryContoursCheckBoxValueChanged, true);
8527            app.PlotBathymetryContoursCheckBox.Enable = 'off';
8528            app.PlotBathymetryContoursCheckBox.Text = 'Plot Bathymetry
Contours';
8529            app.PlotBathymetryContoursCheckBox.FontWeight = 'bold';
8530            app.PlotBathymetryContoursCheckBox.Position = [45 141 170 22];
8531
8532            % Create MinimumEditField_2Label
8533            app.MinimumEditField_2Label = uilabel(app.
BathymetryContoursTab_2);
8534            app.MinimumEditField_2Label.Position = [45 86 55 22];
8535            app.MinimumEditField_2Label.Text = 'Minimum';
8536
8537            % Create MinimumEditField_2
8538            app.MinimumEditField_2 = uieditfield(app.BathymetryContoursTab_2,
'numeric');
8539            app.MinimumEditField_2.Limits = [0 Inf];
8540            app.MinimumEditField_2.ValueDisplayFormat = '%8.1f';
8541            app.MinimumEditField_2.ValueChangedFcn = createCallbackFcn(app,
@MinimumEditField_2ValueChanged, true);
8542            app.MinimumEditField_2.FontColor = [0.651 0.651 0.651];
8543            app.MinimumEditField_2.Enable = 'off';
8544            app.MinimumEditField_2.Tooltip = {'0 value = coastline'; '';
'Lower values indicate shallower waters.'};
8545            app.MinimumEditField_2.Position = [102 89 57 16];
8546
8547            % Create MaximumEditField_2Label
8548            app.MaximumEditField_2Label = uilabel(app.
BathymetryContoursTab_2);
8549            app.MaximumEditField_2Label.Position = [45 32 58 22];
8550            app.MaximumEditField_2Label.Text = 'Maximum';
8551
8552            % Create MaximumEditField_2
8553            app.MaximumEditField_2 = uieditfield(app.BathymetryContoursTab_2,
'numeric');
8554            app.MaximumEditField_2.Limits = [0.1 Inf];
8555            app.MaximumEditField_2.ValueDisplayFormat = '%8.1f';
8556            app.MaximumEditField_2.ValueChangedFcn = createCallbackFcn(app,
@MaximumEditField_2ValueChanged, true);
8557            app.MaximumEditField_2.FontColor = [0.651 0.651 0.651];
8558            app.MaximumEditField_2.Enable = 'off';
8559            app.MaximumEditField_2.Tooltip = {'Higher value indicate deeper
```

```
waters'};
8560             app.MaximumEditField_2.Position = [102 35 57 16];
8561             app.MaximumEditField_2.Value = 10000;
8562
8563             % Create IntervalEditField_6Label
8564             app.IntervalEditField_6Label = uilabel(app.↙
BathymetryContoursTab_2);
8565             app.IntervalEditField_6Label.Position = [45 59 45 22];
8566             app.IntervalEditField_6Label.Text = 'Interval';
8567
8568             % Create IntervalEditField_6
8569             app.IntervalEditField_6 = uieditfield(app.BathymetryContoursTab_2,↙
'numeric');
8570             app.IntervalEditField_6.Limits = [0 Inf];
8571             app.IntervalEditField_6.ValueDisplayFormat = '%8.1f';
8572             app.IntervalEditField_6.ValueChangedFcn = createCallbackFcn(app,↙
@IntervalEditField_6ValueChanged, true);
8573             app.IntervalEditField_6.FontColor = [0.651 0.651 0.651];
8574             app.IntervalEditField_6.Enable = 'off';
8575             app.IntervalEditField_6.Tooltip = {'Interval between adjacent↙
contours'};
8576             app.IntervalEditField_6.Position = [102 62 48 16];
8577             app.IntervalEditField_6.Value = 500;
8578
8579             % Create LineLabel_4
8580             app.LineLabel_4 = uilabel(app.BathymetryContoursTab_2);
8581             app.LineLabel_4.FontWeight = 'bold';
8582             app.LineLabel_4.Position = [196 109 30 22];
8583             app.LineLabel_4.Text = 'Line';
8584
8585             % Create AddLabelCheckBox_2
8586             app.AddLabelCheckBox_2 = uicheckbox(app.BathymetryContoursTab_2);
8587             app.AddLabelCheckBox_2.ValueChangedFcn = createCallbackFcn(app,↙
@AddLabelCheckBox_2ValueChanged, true);
8588             app.AddLabelCheckBox_2.Enable = 'off';
8589             app.AddLabelCheckBox_2.Text = 'Add Label';
8590             app.AddLabelCheckBox_2.FontWeight = 'bold';
8591             app.AddLabelCheckBox_2.Position = [375 109 80 22];
8592
8593             % Create ColorDropDown_4Label
8594             app.ColorDropDown_4Label = uilabel(app.BathymetryContoursTab_2);
8595             app.ColorDropDown_4Label.Position = [196 32 35 22];
8596             app.ColorDropDown_4Label.Text = 'Color';
8597
8598             % Create ColorDropDown_4
8599             app.ColorDropDown_4 = uidropdown(app.BathymetryContoursTab_2);
8600             app.ColorDropDown_4.Items = {'Black', 'Dark gray', 'Medium gray',↙
'Light gray', 'Red', 'Green', 'Blue', 'Yellow', 'Cyan', 'Magenta', 'White'};
8601             app.ColorDropDown_4.Enable = 'off';
8602             app.ColorDropDown_4.Position = [234 35 104 16];
8603             app.ColorDropDown_4.Value = 'Black';
8604
8605             % Create StyleDropDown_3Label_2
8606             app.StyleDropDown_3Label_2 = uilabel(app.BathymetryContoursTab_2);
8607             app.StyleDropDown_3Label_2.Position = [196 59 32 22];
8608             app.StyleDropDown_3Label_2.Text = 'Style';
8609
8610             % Create StyleDropDown_3
```

```
8611            app.StyleDropDown_3 = uidropdown(app.BathymetryContoursTab_2);
8612            app.StyleDropDown_3.Items = {'Solid', 'Dashed', 'Dotted', 'Dash-dotted'};
8613            app.StyleDropDown_3.Enable = 'off';
8614            app.StyleDropDown_3.Position = [234 62 104 16];
8615            app.StyleDropDown_3.Value = 'Solid';
8616
8617            % Create WidthEditField_2Label
8618            app.WidthEditField_2Label = uilabel(app.BathymetryContoursTab_2);
8619            app.WidthEditField_2Label.Position = [196 86 37 22];
8620            app.WidthEditField_2Label.Text = 'Width';
8621
8622            % Create WidthEditField_2
8623            app.WidthEditField_2 = uieditfield(app.BathymetryContoursTab_2, 'numeric');
8624            app.WidthEditField_2.Limits = [0 Inf];
8625            app.WidthEditField_2.ValueDisplayFormat = '%3.1f';
8626            app.WidthEditField_2.ValueChangedFcn = createCallbackFcn(app, @WidthEditField_2ValueChanged, true);
8627            app.WidthEditField_2.FontColor = [0.651 0.651 0.651];
8628            app.WidthEditField_2.Enable = 'off';
8629            app.WidthEditField_2.Tooltip = {'Line thickness'};
8630            app.WidthEditField_2.Position = [234 89 33 16];
8631            app.WidthEditField_2.Value = 0.1;
8632
8633            % Create IntervalEditField_8Label
8634            app.IntervalEditField_8Label = uilabel(app.BathymetryContoursTab_2);
8635            app.IntervalEditField_8Label.Tooltip = {'Add labels at specified intervals.'; ''; 'Suggestion: Set at intervals divisible by the contour range interval'; ''; ''; 'Suggestion: Set at intervals divisible by the depth range interval'};
8636            app.IntervalEditField_8Label.Position = [375 59 45 22];
8637            app.IntervalEditField_8Label.Text = 'Interval';
8638
8639            % Create LabelSizeCont
8640            app.LabelSizeCont = uieditfield(app.BathymetryContoursTab_2, 'numeric');
8641            app.LabelSizeCont.Limits = [0 30];
8642            app.LabelSizeCont.ValueDisplayFormat = '%8.0f';
8643            app.LabelSizeCont.ValueChangedFcn = createCallbackFcn(app, @LabelSizeContValueChanged, true);
8644            app.LabelSizeCont.FontColor = [0.651 0.651 0.651];
8645            app.LabelSizeCont.Enable = 'off';
8646            app.LabelSizeCont.Tooltip = {''};
8647            app.LabelSizeCont.Position = [426 89 40 16];
8648            app.LabelSizeCont.Value = 8;
8649
8650            % Create SpacingEditField_4Label_3
8651            app.SpacingEditField_4Label_3 = uilabel(app.BathymetryContoursTab_2);
8652            app.SpacingEditField_4Label_3.Tooltip = {'Font size'};
8653            app.SpacingEditField_4Label_3.Position = [375 86 28 22];
8654            app.SpacingEditField_4Label_3.Text = 'Size';
8655
8656            % Create SpacingEditField_4Label_2
8657            app.SpacingEditField_4Label_2 = uilabel(app.BathymetryContoursTab_2);
```

```
8658            app.SpacingEditField_4Label_2.WordWrap = 'on';
8659            app.SpacingEditField_4Label_2.Tooltip = {'Adjust the distance between the labels'; ''; 'Higher values reduce label crowding'};
8660            app.SpacingEditField_4Label_2.Position = [375 29 77 29];
8661            app.SpacingEditField_4Label_2.Text = 'Spacing';
8662
8663            % Create SpacingEditField
8664            app.SpacingEditField = uieditfield(app.BathymetryContoursTab_2, 'numeric');
8665            app.SpacingEditField.Limits = [0 Inf];
8666            app.SpacingEditField.ValueDisplayFormat = '%8.1f';
8667            app.SpacingEditField.ValueChangedFcn = createCallbackFcn(app, @SpacingEditFieldValueChanged, true);
8668            app.SpacingEditField.FontColor = [0.651 0.651 0.651];
8669            app.SpacingEditField.Enable = 'off';
8670            app.SpacingEditField.Tooltip = {''};
8671            app.SpacingEditField.Position = [426 35 47 16];
8672            app.SpacingEditField.Value = 100;
8673
8674            % Create IntervalEditField_8
8675            app.IntervalEditField_8 = uieditfield(app.BathymetryContoursTab_2, 'numeric');
8676            app.IntervalEditField_8.Limits = [0 Inf];
8677            app.IntervalEditField_8.ValueDisplayFormat = '%8.1f';
8678            app.IntervalEditField_8.ValueChangedFcn = createCallbackFcn(app, @IntervalEditField_8ValueChanged, true);
8679            app.IntervalEditField_8.FontColor = [0.651 0.651 0.651];
8680            app.IntervalEditField_8.Enable = 'off';
8681            app.IntervalEditField_8.Tooltip = {''};
8682            app.IntervalEditField_8.Position = [426 62 47 16];
8683            app.IntervalEditField_8.Value = 100;
8684
8685            % Create GaugesTab_2
8686            app.GaugesTab_2 = uitab(app.TabGroup3);
8687            app.GaugesTab_2.AutoResizeChildren = 'off';
8688            app.GaugesTab_2.SizeChangedFcn = createCallbackFcn(app, @GaugesTab_2SizeChanged, true);
8689            app.GaugesTab_2.Title = 'Gauges';
8690
8691            % Create CoastlinecolourDropDownLabel_19
8692            app.CoastlinecolourDropDownLabel_19 = uilabel(app.GaugesTab_2);
8693            app.CoastlinecolourDropDownLabel_19.Position = [45 57 32 22];
8694            app.CoastlinecolourDropDownLabel_19.Text = 'Style';
8695
8696            % Create gaugemarkerVelocityTab
8697            app.gaugemarkerVelocityTab = uidropdown(app.GaugesTab_2);
8698            app.gaugemarkerVelocityTab.Items = {'o', '+', '*', '.', 'x', '-', '|', '^', 'v', '>', '<', 'diamond', 'hexagram', 'pentagram', 'square'};
8699            app.gaugemarkerVelocityTab.Enable = 'off';
8700            app.gaugemarkerVelocityTab.Tooltip = {'Marker style'};
8701            app.gaugemarkerVelocityTab.Position = [81 60 92 16];
8702            app.gaugemarkerVelocityTab.Value = 'o';
8703
8704            % Create ColorDropDown_7Label
8705            app.ColorDropDown_7Label = uilabel(app.GaugesTab_2);
8706            app.ColorDropDown_7Label.Position = [45 30 35 22];
8707            app.ColorDropDown_7Label.Text = 'Color';
8708
```

```
8709            % Create ColorDropDown_7
8710            app.ColorDropDown_7 = uidropdown(app.GaugesTab_2);
8711            app.ColorDropDown_7.Items = {'Black', 'Dark gray', 'Medium gray',↙
'Light gray', 'Red', 'Green', 'Blue', 'Yellow', 'Cyan', 'Magenta', 'White'};
8712            app.ColorDropDown_7.Enable = 'off';
8713            app.ColorDropDown_7.Position = [81 33 103 16];
8714            app.ColorDropDown_7.Value = 'Black';
8715
8716            % Create Button_27
8717            app.Button_27 = uibutton(app.GaugesTab_2, 'push');
8718            app.Button_27.ButtonPushedFcn = createCallbackFcn(app,↙
@Button_27Pushed, true);
8719            app.Button_27.Enable = 'off';
8720            app.Button_27.Tooltip = {'Load the file'; ''; 'File extension must↙
be .shp or .txt (tab delimited)'};
8721            app.Button_27.Position = [211 116 19 19];
8722            app.Button_27.Text = '...';
8723
8724            % Create GaugeFileLabel_2
8725            app.GaugeFileLabel_2 = uilabel(app.GaugesTab_2);
8726            app.GaugeFileLabel_2.FontWeight = 'bold';
8727            app.GaugeFileLabel_2.Position = [45 114 26 22];
8728            app.GaugeFileLabel_2.Text = 'File';
8729
8730            % Create FileEditField_2
8731            app.FileEditField_2 = uieditfield(app.GaugesTab_2, 'text');
8732            app.FileEditField_2.ValueChangedFcn = createCallbackFcn(app,↙
@FileEditField_2ValueChanged, true);
8733            app.FileEditField_2.Editable = 'off';
8734            app.FileEditField_2.Enable = 'off';
8735            app.FileEditField_2.Tooltip = {'File extension must be .shp or .↙
txt'};
8736            app.FileEditField_2.Placeholder = '.shp or .txt';
8737            app.FileEditField_2.Position = [74 117 130 16];
8738
8739            % Create PlotGaugesCheckBox
8740            app.PlotGaugesCheckBox = uicheckbox(app.GaugesTab_2);
8741            app.PlotGaugesCheckBox.ValueChangedFcn = createCallbackFcn(app,↙
@PlotGaugesCheckBoxValueChanged, true);
8742            app.PlotGaugesCheckBox.Text = 'Plot Gauges';
8743            app.PlotGaugesCheckBox.FontWeight = 'bold';
8744            app.PlotGaugesCheckBox.Position = [45 141 92 22];
8745
8746            % Create AddLabelCheckBox
8747            app.AddLabelCheckBox = uicheckbox(app.GaugesTab_2);
8748            app.AddLabelCheckBox.ValueChangedFcn = createCallbackFcn(app,↙
@AddLabelCheckBoxValueChanged, true);
8749            app.AddLabelCheckBox.Enable = 'off';
8750            app.AddLabelCheckBox.Tooltip = {'The label follows the sequence of↙
latitude-longitude combinations from the loaded file.'; ''; 'The labels correspond↙
to the numeric values in the sta_ files.'};
8751            app.AddLabelCheckBox.Text = '  Add Label';
8752            app.AddLabelCheckBox.FontWeight = 'bold';
8753            app.AddLabelCheckBox.Position = [254 84 86 22];
8754
8755            % Create FontSizeEditField_2Label
8756            app.FontSizeEditField_2Label = uilabel(app.GaugesTab_2);
8757            app.FontSizeEditField_2Label.Enable = 'off';
```

```
8758            app.FontSizeEditField_2Label.Tooltip = {'Font size'};
8759            app.FontSizeEditField_2Label.Position = [255 57 33 22];
8760            app.FontSizeEditField_2Label.Text = 'Size';
8761
8762            % Create FontSizeEditField_2
8763            app.FontSizeEditField_2 = uieditfield(app.GaugesTab_2, 'numeric');
8764            app.FontSizeEditField_2.Limits = [0 100];
8765            app.FontSizeEditField_2.ValueDisplayFormat = '%8.0f';
8766            app.FontSizeEditField_2.ValueChangedFcn = createCallbackFcn(app,↙
@FontSizeEditField_2ValueChanged, true);
8767            app.FontSizeEditField_2.FontColor = [0.651 0.651 0.651];
8768            app.FontSizeEditField_2.Enable = 'off';
8769            app.FontSizeEditField_2.Tooltip = {'Font size'};
8770            app.FontSizeEditField_2.Position = [290 60 30 16];
8771            app.FontSizeEditField_2.Value = 12;
8772
8773            % Create GLspacing_2
8774            app.GLspacing_2 = uieditfield(app.GaugesTab_2, 'numeric');
8775            app.GLspacing_2.Limits = [0 100];
8776            app.GLspacing_2.ValueDisplayFormat = '%8.3f';
8777            app.GLspacing_2.ValueChangedFcn = createCallbackFcn(app,↙
@GLspacing_2ValueChanged, true);
8778            app.GLspacing_2.FontColor = [0.651 0.651 0.651];
8779            app.GLspacing_2.Enable = 'off';
8780            app.GLspacing_2.Tooltip = {''};
8781            app.GLspacing_2.Position = [394 60 40 16];
8782            app.GLspacing_2.Value = 0.5;
8783
8784            % Create GLspacingLabel_2
8785            app.GLspacingLabel_2 = uilabel(app.GaugesTab_2);
8786            app.GLspacingLabel_2.Enable = 'off';
8787            app.GLspacingLabel_2.Tooltip = {'Horizontal distance between the↙
point and the label. '; ' '; ' A value of 0.5 means that a label for a point at↙
120°E will be placed at 120.5°E.'; ''; 'Unit: degrees / meters'};
8788            app.GLspacingLabel_2.Position = [339 57 50 22];
8789            app.GLspacingLabel_2.Text = 'Spacing';
8790
8791            % Create AlignmentLabel_2
8792            app.AlignmentLabel_2 = uilabel(app.GaugesTab_2);
8793            app.AlignmentLabel_2.Enable = 'off';
8794            app.AlignmentLabel_2.Tooltip = {'Horizontal and vertical↙
alignments relative to the marker location'};
8795            app.AlignmentLabel_2.Position = [255 30 59 22];
8796            app.AlignmentLabel_2.Text = 'Alignment';
8797
8798            % Create CoastlinecolourDropDown_10
8799            app.CoastlinecolourDropDown_10 = uidropdown(app.GaugesTab_2);
8800            app.CoastlinecolourDropDown_10.Items = {'Centre', 'Right',↙
'Left'};
8801            app.CoastlinecolourDropDown_10.Enable = 'off';
8802            app.CoastlinecolourDropDown_10.Tooltip = {''};
8803            app.CoastlinecolourDropDown_10.Position = [314 33 70 16];
8804            app.CoastlinecolourDropDown_10.Value = 'Right';
8805
8806            % Create CoastlinecolourDropDown_11
8807            app.CoastlinecolourDropDown_11 = uidropdown(app.GaugesTab_2);
8808            app.CoastlinecolourDropDown_11.Items = {'Centre', 'Top',↙
'Bottom'};
```

```
8809            app.CoastlinecolourDropDown_11.Enable = 'off';
8810            app.CoastlinecolourDropDown_11.Tooltip = {''};
8811            app.CoastlinecolourDropDown_11.Position = [389 33 70 16];
8812            app.CoastlinecolourDropDown_11.Value = 'Centre';
8813
8814            % Create SizeEditField_2Label
8815            app.SizeEditField_2Label = uilabel(app.GaugesTab_2);
8816            app.SizeEditField_2Label.Position = [45 84 28 22];
8817            app.SizeEditField_2Label.Text = 'Size';
8818
8819            % Create SizeEditField_2
8820            app.SizeEditField_2 = uieditfield(app.GaugesTab_2, 'numeric');
8821            app.SizeEditField_2.Limits = [0 100];
8822            app.SizeEditField_2.ValueDisplayFormat = '%3.0f';
8823            app.SizeEditField_2.ValueChangedFcn = createCallbackFcn(app, ↙
@SizeEditField_2ValueChanged, true);
8824            app.SizeEditField_2.FontColor = [0.651 0.651 0.651];
8825            app.SizeEditField_2.Enable = 'off';
8826            app.SizeEditField_2.Position = [81 87 30 16];
8827            app.SizeEditField_2.Value = 5;
8828
8829            % Create Panel_17
8830            app.Panel_17 = uipanel(app.VelocityMapTab);
8831            app.Panel_17.AutoResizeChildren = 'off';
8832            app.Panel_17.Position = [11 39 520 77];
8833
8834            % Create pngCheckBox_3
8835            app.pngCheckBox_3 = uicheckbox(app.Panel_17);
8836            app.pngCheckBox_3.Tooltip = {'Resolution is set to 300 DPI'};
8837            app.pngCheckBox_3.Text = '.png';
8838            app.pngCheckBox_3.Position = [186 12 46 22];
8839
8840            % Create FileFormatLabel_3
8841            app.FileFormatLabel_3 = uilabel(app.Panel_17);
8842            app.FileFormatLabel_3.Position = [69 12 66 22];
8843            app.FileFormatLabel_3.Text = 'File Format';
8844
8845            % Create jpgCheckBox_3
8846            app.jpgCheckBox_3 = uicheckbox(app.Panel_17);
8847            app.jpgCheckBox_3.Tooltip = {'Resolution is set to 300 DPI'};
8848            app.jpgCheckBox_3.Text = '.jpg';
8849            app.jpgCheckBox_3.Position = [138 12 42 22];
8850
8851            % Create Button_26
8852            app.Button_26 = uibutton(app.Panel_17, 'push');
8853            app.Button_26.ButtonPushedFcn = createCallbackFcn(app, ↙
@Button_26Pushed, true);
8854            app.Button_26.FontSize = 10;
8855            app.Button_26.FontAngle = 'italic';
8856            app.Button_26.Tooltip = {'Load the preferred directory.'; ''; ↙
'"OUTPUT_FILES/Figures"  folder will be created in the selected directory.'};
8857            app.Button_26.Position = [403 42 19 19];
8858            app.Button_26.Text = '...';
8859
8860            % Create FramerateEditFieldLabel
8861            app.FramerateEditFieldLabel = uilabel(app.Panel_17);
8862            app.FramerateEditFieldLabel.WordWrap = 'on';
8863            app.FramerateEditFieldLabel.Visible = 'off';
```

```
8864            app.FramerateEditFieldLabel.Tooltip = {'Frames per second.'; ''; 'Lower value = slower animation.'};
8865            app.FramerateEditFieldLabel.Position = [383 7 77 33];
8866            app.FramerateEditFieldLabel.Text = 'Frame rate';
8867
8868            % Create FramerateEditField
8869            app.FramerateEditField = uieditfield(app.Panel_17, 'numeric');
8870            app.FramerateEditField.Limits = [0 100];
8871            app.FramerateEditField.ValueDisplayFormat = '%.0f';
8872            app.FramerateEditField.ValueChangedFcn = createCallbackFcn(app, @FramerateEditFieldValueChanged, true);
8873            app.FramerateEditField.FontColor = [0.651 0.651 0.651];
8874            app.FramerateEditField.Visible = 'off';
8875            app.FramerateEditField.Tooltip = {''};
8876            app.FramerateEditField.Position = [448 15 30 17];
8877            app.FramerateEditField.Value = 5;
8878
8879            % Create mp4CheckBox_2
8880            app.mp4CheckBox_2 = uicheckbox(app.Panel_17);
8881            app.mp4CheckBox_2.ValueChangedFcn = createCallbackFcn(app, @mp4CheckBox_2ValueChanged, true);
8882            app.mp4CheckBox_2.Enable = 'off';
8883            app.mp4CheckBox_2.Tooltip = {'Create an animation.'; ''; 'File is saved as "animation.mp4."'; ''; 'Enabled when "Plot separately" is selected.'};
8884            app.mp4CheckBox_2.Text = '.mp4';
8885            app.mp4CheckBox_2.Position = [324 12 49 22];
8886
8887            % Create OutputDirectoryLabel_2
8888            app.OutputDirectoryLabel_2 = uilabel(app.Panel_17);
8889            app.OutputDirectoryLabel_2.Tooltip = {'Load the preferred directory.'; ''; '''OUTPUT_FILES/Figures'' folder will be created in the selected directory.'};
8890            app.OutputDirectoryLabel_2.Position = [40 40 94 22];
8891            app.OutputDirectoryLabel_2.Text = 'Output Directory';
8892
8893            % Create OutputDirectoryEditField_3
8894            app.OutputDirectoryEditField_3 = uieditfield(app.Panel_17, 'text');
8895            app.OutputDirectoryEditField_3.ValueChangedFcn = createCallbackFcn(app, @OutputDirectoryEditField_3ValueChanged, true);
8896            app.OutputDirectoryEditField_3.Editable = 'off';
8897            app.OutputDirectoryEditField_3.FontColor = [0 0 1];
8898            app.OutputDirectoryEditField_3.Tooltip = {''};
8899            app.OutputDirectoryEditField_3.Placeholder = 'Default: Desktop';
8900            app.OutputDirectoryEditField_3.Position = [139 43 257 16];
8901
8902            % Create txtCheckBox_2
8903            app.txtCheckBox_2 = uicheckbox(app.Panel_17);
8904            app.txtCheckBox_2.Enable = 'off';
8905            app.txtCheckBox_2.Tooltip = {'Save raw data of U and V vector files as tab-delimited texts.'; ''; 'It is enabled when "Plot separately" is selected.'};
8906            app.txtCheckBox_2.Text = '.txt';
8907            app.txtCheckBox_2.Position = [238 12 39 22];
8908
8909            % Create tifCheckBox
8910            app.tifCheckBox = uicheckbox(app.Panel_17);
8911            app.tifCheckBox.Enable = 'off';
```

```
8912            app.tifCheckBox.Tooltip = {'Georeferenced raster file.'; ''; ↙
'Enabled when "Plot separately" is selected.'};
8913            app.tifCheckBox.Text = '.tif';
8914            app.tifCheckBox.Position = [283 12 35 22];
8915
8916            % Create SaveMapLabel_2
8917            app.SaveMapLabel_2 = uilabel(app.VelocityMapTab);
8918            app.SaveMapLabel_2.BackgroundColor = [0.9412 0.9412 0.9412];
8919            app.SaveMapLabel_2.FontSize = 15;
8920            app.SaveMapLabel_2.FontWeight = 'bold';
8921            app.SaveMapLabel_2.FontColor = [0.0314 0.3686 0.6];
8922            app.SaveMapLabel_2.Position = [38 105 92 22];
8923            app.SaveMapLabel_2.Text = '  Save Map  ';
8924
8925            % Create BasemapandOverlaysLabel
8926            app.BasemapandOverlaysLabel = uilabel(app.VelocityMapTab);
8927            app.BasemapandOverlaysLabel.BackgroundColor = [0.9412 0.9412 ↙
0.9412];
8928            app.BasemapandOverlaysLabel.FontSize = 15;
8929            app.BasemapandOverlaysLabel.FontWeight = 'bold';
8930            app.BasemapandOverlaysLabel.FontColor = [0.0314 0.3686 0.6];
8931            app.BasemapandOverlaysLabel.Position = [38 527 187 22];
8932            app.BasemapandOverlaysLabel.Text = '  Basemap and Overlays  ';
8933
8934            % Create Panel_18
8935            app.Panel_18 = uipanel(app.VelocityMapTab);
8936            app.Panel_18.AutoResizeChildren = 'off';
8937            app.Panel_18.Position = [11 128 520 172];
8938
8939            % Create BoundaryLimitsLabel_2
8940            app.BoundaryLimitsLabel_2 = uilabel(app.Panel_18);
8941            app.BoundaryLimitsLabel_2.FontWeight = 'bold';
8942            app.BoundaryLimitsLabel_2.Tooltip = {'When all textboxes are left ↙
unchanged (all zeroes), the values are automatically set based on the extent of the ↙
uploaded file and the input southwest corner coordinates'};
8943            app.BoundaryLimitsLabel_2.Position = [25 91 99 22];
8944            app.BoundaryLimitsLabel_2.Text = 'Boundary Limits';
8945
8946            % Create PlotindegreesCheckBox
8947            app.PlotindegreesCheckBox = uicheckbox(app.Panel_18);
8948            app.PlotindegreesCheckBox.ValueChangedFcn = createCallbackFcn(app, ↙
@PlotindegreesCheckBoxValueChanged, true);
8949            app.PlotindegreesCheckBox.Tooltip = {'Change the label format into ↙
geographic coordinates'};
8950            app.PlotindegreesCheckBox.Text = 'Plot in degrees';
8951            app.PlotindegreesCheckBox.FontSize = 11;
8952            app.PlotindegreesCheckBox.Position = [403 56 96 22];
8953
8954            % Create EastEditField_2Label
8955            app.EastEditField_2Label = uilabel(app.Panel_18);
8956            app.EastEditField_2Label.Position = [331 62 29 22];
8957            app.EastEditField_2Label.Text = 'East';
8958
8959            % Create EastEditField_2
8960            app.EastEditField_2 = uieditfield(app.Panel_18, 'numeric');
8961            app.EastEditField_2.ValueDisplayFormat = '%8.4f';
8962            app.EastEditField_2.ValueChangedFcn = createCallbackFcn(app, ↙
@EastEditField_2ValueChanged, true);
```

```
8963            app.EastEditField_2.FontColor = [0.651 0.651 0.651];
8964            app.EastEditField_2.Position = [265 65 60 16];
8965
8966            % Create WestEditField_2Label
8967            app.WestEditField_2Label = uilabel(app.Panel_18);
8968            app.WestEditField_2Label.HorizontalAlignment = 'right';
8969            app.WestEditField_2Label.Position = [87 62 32 22];
8970            app.WestEditField_2Label.Text = 'West';
8971
8972            % Create WestEditField_2
8973            app.WestEditField_2 = uieditfield(app.Panel_18, 'numeric');
8974            app.WestEditField_2.ValueDisplayFormat = '%8.4f';
8975            app.WestEditField_2.ValueChangedFcn = createCallbackFcn(app, ↙
@WestEditField_2ValueChanged, true);
8976            app.WestEditField_2.FontColor = [0.651 0.651 0.651];
8977            app.WestEditField_2.Position = [125 65 60 16];
8978
8979            % Create NorthEditField_2Label
8980            app.NorthEditField_2Label = uilabel(app.Panel_18);
8981            app.NorthEditField_2Label.HorizontalAlignment = 'right';
8982            app.NorthEditField_2Label.Position = [206 91 35 22];
8983            app.NorthEditField_2Label.Text = 'North';
8984
8985            % Create NorthEditField_2
8986            app.NorthEditField_2 = uieditfield(app.Panel_18, 'numeric');
8987            app.NorthEditField_2.ValueDisplayFormat = '%8.4f';
8988            app.NorthEditField_2.ValueChangedFcn = createCallbackFcn(app, ↙
@NorthEditField_2ValueChanged, true);
8989            app.NorthEditField_2.FontColor = [0.651 0.651 0.651];
8990            app.NorthEditField_2.Position = [195 77 60 16];
8991
8992            % Create SouthEditField_2Label
8993            app.SouthEditField_2Label = uilabel(app.Panel_18);
8994            app.SouthEditField_2Label.HorizontalAlignment = 'right';
8995            app.SouthEditField_2Label.Position = [205 34 37 22];
8996            app.SouthEditField_2Label.Text = 'South';
8997
8998            % Create SouthEditField_2
8999            app.SouthEditField_2 = uieditfield(app.Panel_18, 'numeric');
9000            app.SouthEditField_2.ValueDisplayFormat = '%8.4f';
9001            app.SouthEditField_2.ValueChangedFcn = createCallbackFcn(app, ↙
@SouthEditField_2ValueChanged, true);
9002            app.SouthEditField_2.FontColor = [0.651 0.651 0.651];
9003            app.SouthEditField_2.Position = [195 53 60 16];
9004
9005            % Create Height_3
9006            app.Height_3 = uieditfield(app.Panel_18, 'numeric');
9007            app.Height_3.Limits = [1 Inf];
9008            app.Height_3.ValueDisplayFormat = '%2.0f';
9009            app.Height_3.ValueChangedFcn = createCallbackFcn(app, ↙
@Height_3ValueChanged, true);
9010            app.Height_3.FontColor = [0.651 0.651 0.651];
9011            app.Height_3.Enable = 'off';
9012            app.Height_3.Tooltip = {''};
9013            app.Height_3.Position = [330 12 40 16];
9014            app.Height_3.Value = 8;
9015
9016            % Create ThicknessLabel_10
```

```
9017             app.ThicknessLabel_10 = uilabel(app.Panel_18);
9018             app.ThicknessLabel_10.Tooltip = {'Unit: inches'};
9019             app.ThicknessLabel_10.Position = [283 9 41 22];
9020             app.ThicknessLabel_10.Text = 'Height';
9021 
9022             % Create AutoSetCheckBox_3
9023             app.AutoSetCheckBox_3 = uicheckbox(app.Panel_18);
9024             app.AutoSetCheckBox_3.ValueChangedFcn = createCallbackFcn(app, ↙
@AutoSetCheckBox_3ValueChanged, true);
9025             app.AutoSetCheckBox_3.Text = ' Auto Set';
9026             app.AutoSetCheckBox_3.FontSize = 11;
9027             app.AutoSetCheckBox_3.Position = [102 9 68 22];
9028             app.AutoSetCheckBox_3.Value = true;
9029 
9030             % Create Width_3
9031             app.Width_3 = uieditfield(app.Panel_18, 'numeric');
9032             app.Width_3.Limits = [1 Inf];
9033             app.Width_3.ValueDisplayFormat = '%2.0f';
9034             app.Width_3.ValueChangedFcn = createCallbackFcn(app, ↙
@Width_3ValueChanged, true);
9035             app.Width_3.FontColor = [0.651 0.651 0.651];
9036             app.Width_3.Enable = 'off';
9037             app.Width_3.Tooltip = {''};
9038             app.Width_3.Position = [225 12 41 16];
9039             app.Width_3.Value = 11;
9040 
9041             % Create ThicknessLabel_11
9042             app.ThicknessLabel_11 = uilabel(app.Panel_18);
9043             app.ThicknessLabel_11.Tooltip = {'Unit: inches'};
9044             app.ThicknessLabel_11.Position = [188 9 37 22];
9045             app.ThicknessLabel_11.Text = 'Width';
9046 
9047             % Create FigureSizeLabel_3
9048             app.FigureSizeLabel_3 = uilabel(app.Panel_18);
9049             app.FigureSizeLabel_3.FontWeight = 'bold';
9050             app.FigureSizeLabel_3.Position = [25 9 69 22];
9051             app.FigureSizeLabel_3.Text = 'Figure Size';
9052 
9053             % Create CloseFiguresButton_3
9054             app.CloseFiguresButton_3 = uibutton(app.Panel_18, 'push');
9055             app.CloseFiguresButton_3.ButtonPushedFcn = createCallbackFcn(app, ↙
@CloseFiguresButton_3Pushed, true);
9056             app.CloseFiguresButton_3.Tooltip = {''};
9057             app.CloseFiguresButton_3.Position = [403 10 89 20];
9058             app.CloseFiguresButton_3.Text = 'Close Figures';
9059 
9060             % Create SettoDefaultButton
9061             app.SettoDefaultButton = uibutton(app.Panel_18, 'push');
9062             app.SettoDefaultButton.ButtonPushedFcn = createCallbackFcn(app, ↙
@SettoDefaultButtonPushed, true);
9063             app.SettoDefaultButton.Tooltip = {'Use the boundary limits of the↙
U-vector file'};
9064             app.SettoDefaultButton.Position = [403 76 84 20];
9065             app.SettoDefaultButton.Text = 'Set to Default';
9066 
9067             % Create ButtonGroup_12
9068             app.ButtonGroup_12 = uibuttongroup(app.Panel_18);
9069             app.ButtonGroup_12.AutoResizeChildren = 'off';
```

```
9070             app.ButtonGroup_12.SelectionChangedFcn = createCallbackFcn(app, ↙
@ButtonGroup_12SelectionChanged, true);
9071             app.ButtonGroup_12.SizeChangedFcn = createCallbackFcn(app, ↙
@ButtonGroup_12SizeChanged, true);
9072             app.ButtonGroup_12.Position = [106 126 311 30];
9073 
9074             % Create PlotalldatainonefigureButton_2
9075             app.PlotalldatainonefigureButton_2 = uiradiobutton(app. ↙
ButtonGroup_12);
9076             app.PlotalldatainonefigureButton_2.Tooltip = {'All maps are↙
displayed as subplots in one window'};
9077             app.PlotalldatainonefigureButton_2.Text = 'Plot all data in one↙
figure';
9078             app.PlotalldatainonefigureButton_2.FontWeight = 'bold';
9079             app.PlotalldatainonefigureButton_2.Position = [13 4 164 22];
9080             app.PlotalldatainonefigureButton_2.Value = true;
9081 
9082             % Create PlotseparatelyButton_3
9083             app.PlotseparatelyButton_3 = uiradiobutton(app.ButtonGroup_12);
9084             app.PlotseparatelyButton_3.Tooltip = {'One map per window'};
9085             app.PlotseparatelyButton_3.Text = 'Plot separately';
9086             app.PlotseparatelyButton_3.FontWeight = 'bold';
9087             app.PlotseparatelyButton_3.Position = [193 4 108 22];
9088 
9089             % Create GeneralLayoutLabel_2
9090             app.GeneralLayoutLabel_2 = uilabel(app.VelocityMapTab);
9091             app.GeneralLayoutLabel_2.BackgroundColor = [0.9412 0.9412 0.9412];
9092             app.GeneralLayoutLabel_2.FontSize = 15;
9093             app.GeneralLayoutLabel_2.FontWeight = 'bold';
9094             app.GeneralLayoutLabel_2.FontColor = [0.0314 0.3686 0.6];
9095             app.GeneralLayoutLabel_2.Position = [38 290 131 22];
9096             app.GeneralLayoutLabel_2.Text = '  General Layout  ';
9097 
9098             % Show the figure after all components are created
9099             app.UIFigure.Visible = 'on';
9100         end
9101     end
9102 
9103     % App creation and deletion
9104     methods (Access = public)
9105 
9106         % Construct app
9107         function app = FUNMAP_Output
9108 
9109             runningApp = getRunningApp(app);
9110 
9111             % Check for running singleton app
9112             if isempty(runningApp)
9113 
9114                 % Create UIFigure and components
9115                 createComponents(app)
9116 
9117                 % Register the app with App Designer
9118                 registerApp(app, app.UIFigure)
9119 
9120                 % Execute the startup function
9121                 runStartupFcn(app, @startupFcn)
9122             else
```

```
9123
9124                % Focus the running singleton app
9125                figure(runningApp.UIFigure)
9126
9127                app = runningApp;
9128            end
9129
9130            if nargout == 0
9131                clear app
9132            end
9133        end
9134
9135        % Code that executes before app deletion
9136        function delete(app)
9137
9138            % Delete UIFigure when app is deleted
9139            delete(app.UIFigure)
9140        end
9141    end
9142 end
```