models. As a workaround for not running large models within DeltaShell, it is possible to run the models outside DeltaShell and load their output into DeltaShell. Subsequently D2R can be used to calculate spatiotemporal statistics. This can be done within DeltaShell or in a standalone setup. Subsequently, these spatiotemporal statistics can be used in HABITAT to calculate habitat suitability models. However, in this way the integrated setup in one User Interface is lost.

Conclusions

- The D2R tool was further developed to work within DeltaShell and calculate spatiotemporal statistics from Delwaq output and convert this to unstructured, curvilinear *.asc rasters.
- It was possible to setup a Delft3d-FM, Delwaq and HABITAT model train in DeltaShell
- The Delft3d-FM and Delwaq models could be run in an integrated matter
- Model integration of Delft3D-FM, Delwaq and HABITAT within DeltaShell in its current form is feasible for smaller models in the order of 1000-6000 elements. For larger models, computer clusters are more efficient.

Future developments

- The D2R tool will be expanded to support Delft3D 4, D-Flow FM and D-Waq output (*trim-files and netCDF-files).
- The D-Flow FM and Delwaq results will be converted to HABITAT input with the D2R tool
- The HABITAT model will be run to test and complete the model coupling within DeltaShell

References

Van Oorschot, 2017. Riparian vegetation interacting with river morphology: modelling long-term ecosystem responses to invasive species, climate change, dams and river restoration. PhD thesis.

Van Oorschot et al., 2018. Combined effects of climate change and dam construction on riverine ecosystems. Ecological Engineering 120, pp. 329–344.

Appendix A: Python script to create HABITAT model

The Python script below can be run in HABITAT to create the structure of the HABITAT models for Salmon and Macrophytes. It includes the main models, the submodels and the HSI models, but does not include the input maps. These input maps should be generated with the Delwaq2Raster tool.

```
# This HABITAT model was setup for the TKI project
# "Development of process-based tools for assessing reservoir sediment management"
# It is a test-case to test the integrated DFM morphology - Water Quality and HABITAT setup
within DeltaShell

# Mijke van Oorschot, May 2020

# Disclaimer:
# It is a hypothetical model comprised of 2 species: Macrophytes and Salmon and contains
unvalidated
# response curves to test the calculation of statistics and the integration between the
models.
# These models are not meant to be used other than for testing purposes.

#region imports for habitat functions
from Libraries.StandardFunctions import *
from Libraries.HabitatFunctions import *
import os # for reading paths
import csv # for reading and writing *.csv files

InputDir = "C:\\Users\\oorschot\\OneDrive - Stichting Deltares\\Documents\\Sediment management
Japan\\TKI projects\\2020\Modellen\\HabitatInput\\"

#region setup model structure
compositeModel1 = CreateModel(HabitatModelType.CompositeModel)
compositeModel1.Name = "Macrophytes"

# Create composite model for fetch
compositeModel2 = CreateModel(HabitatModelType.CompositeModel)
compositeModel2.Name = "Salmon"

mainCompositeModel = CreateModel(HabitatModelType.CompositeModel)
mainCompositeModel.Name = "Integrated Habitat Model Test"

# Add sub models (Activities) to main composite model
mainCompositeModel.Activities.Add(compositeModel1)
mainCompositeModel.Activities.Add(compositeModel2)

AddToProject(mainCompositeModel)

# create HSI model for Macrophytes
WaterDepthMeanMac = CreateModel(HabitatModelType.BrokenLinearReclassification)
WaterDepthMeanMac.Name = "WaterDepthMeanMac"
WaterDepthMinMac = CreateModel(HabitatModelType.MultiTableReclassification)
WaterDepthMinMac.Name = "WaterDepthMinMac"
FlowVelocityMaxMac = CreateModel(HabitatModelType.MultiTableReclassification)
FlowVelocityMaxMac.Name = "FlowVelocityMaxMac"
HGI_Macrophytes = CreateModel(HabitatModelType.FormuleBased)
HGI_Macrophytes.Name = "HGI_Macrophytes"
HGI_Macrophytes.Formulas.Clear() # remove output grid

# add models to composite model
compositeModel1.Activities.Add(WaterDepthMeanMac)
compositeModel1.Activities.Add(WaterDepthMinMac)
compositeModel1.Activities.Add(FlowVelocityMaxMac)
compositeModel1.Activities.Add(HGI_Macrophytes)

# create HSI model for Salmon
WaterDepthMeanSal = CreateModel(HabitatModelType.BrokenLinearReclassification)
WaterDepthMeanSal.Name = "WaterDepthMeanSal"
WaterDepthMinSal = CreateModel(HabitatModelType.MultiTableReclassification)
```

```
WaterDepthMinSal.Name = "WaterDepthMinSal"
WaterDepthMaxSal = CreateModel(HabitatModelType.MultiTableReclassification)
WaterDepthMaxSal.Name = "WaterDepthMaxSal"

FlowVelocityMeanSal = CreateModel(HabitatModelType.BrokenLinearReclassification)
FlowVelocityMeanSal.Name = "FlowVelocityMeanSal"
FlowVelocityMinSal = CreateModel(HabitatModelType.MultiTableReclassification)
FlowVelocityMinSal.Name = "FlowVelocityMinSal"
FlowVelocityMaxSal = CreateModel(HabitatModelType.MultiTableReclassification)
FlowVelocityMaxSal.Name = "FlowVelocityMaxSal"
OxygenMinSal = CreateModel(HabitatModelType.BrokenLinearReclassification)
OxygenMinSal.Name = "OxygenMinSal"

HGI_Salmon = CreateModel(HabitatModelType.FormuleBased)
HGI_Salmon.Name = "HGI_Salmon"
HGI_Salmon.Formulas.Clear() # remove output grid

# add models to composite model
compositeModel2.Activities.Add(WaterDepthMeanSal)
compositeModel2.Activities.Add(WaterDepthMinSal)
compositeModel2.Activities.Add(WaterDepthMaxSal)
compositeModel2.Activities.Add(FlowVelocityMeanSal)
compositeModel2.Activities.Add(FlowVelocityMinSal)
compositeModel2.Activities.Add(FlowVelocityMaxSal)
compositeModel2.Activities.Add(OxygenMinSal)
compositeModel2.Activities.Add(HGI_Salmon)
#endregion

# region add response curves (within script)
# Macrophytes response curves
AddBrokenLinearReclassificationRow(WaterDepthMeanMac, 0,0)
AddBrokenLinearReclassificationRow(WaterDepthMeanMac, 0.5,1)
AddBrokenLinearReclassificationRow(WaterDepthMeanMac, 1.5,1)
AddBrokenLinearReclassificationRow(WaterDepthMeanMac, 1.9,0)
AddBrokenLinearReclassificationRow(WaterDepthMeanMac, 10,0)

AddMultiTableReclassificationRow(WaterDepthMinMac, ["[0,0.1]"] , 0, " ")
AddMultiTableReclassificationRow(WaterDepthMinMac, ["<0.1,>"] , 1, " ")

AddMultiTableReclassificationRow(FlowVelocityMaxMac, ["[0,0.35]"] , 1, " ")
AddMultiTableReclassificationRow(FlowVelocityMaxMac, ["<0.35,>"] , 0, " ")

# Salmon response curves
AddBrokenLinearReclassificationRow(FlowVelocityMeanSal, 0,0)
AddBrokenLinearReclassificationRow(FlowVelocityMeanSal, 0.08,0)
AddBrokenLinearReclassificationRow(FlowVelocityMeanSal, 0.13,1)
AddBrokenLinearReclassificationRow(FlowVelocityMeanSal, 0.45,1)
AddBrokenLinearReclassificationRow(FlowVelocityMeanSal, 1,0)
AddBrokenLinearReclassificationRow(FlowVelocityMeanSal, 10,0)

AddMultiTableReclassificationRow(FlowVelocityMaxSal, ["[0,1.0]"] , 1, " ")
AddMultiTableReclassificationRow(FlowVelocityMaxSal, ["<1.0,>"] , 0, " ")

AddMultiTableReclassificationRow(FlowVelocityMinSal, ["[0,0.1]"] , 0, " ")
AddMultiTableReclassificationRow(FlowVelocityMinSal, ["<0.1,>"] , 1, " ")

AddBrokenLinearReclassificationRow(WaterDepthMeanSal, 0,0)
AddBrokenLinearReclassificationRow(WaterDepthMeanSal, 0.15,0)
AddBrokenLinearReclassificationRow(WaterDepthMeanSal, 0.35,1)
AddBrokenLinearReclassificationRow(WaterDepthMeanSal, 0.7,0)
AddBrokenLinearReclassificationRow(WaterDepthMeanSal, 10,0)

AddMultiTableReclassificationRow(WaterDepthMinSal, ["[0,0.15]"] , 0, " ")
AddMultiTableReclassificationRow(WaterDepthMinSal, ["<0.15,>"] , 1, " ")

AddMultiTableReclassificationRow(WaterDepthMaxSal, ["[0,1.5]"] , 1, " ")
AddMultiTableReclassificationRow(WaterDepthMaxSal, ["<1.5,>"] , 0, " ")

AddBrokenLinearReclassificationRow(OxygenMinSal, 0,0)
AddBrokenLinearReclassificationRow(OxygenMinSal, 6,0)
```

```
AddBrokenLinearReclassificationRow(OxygenMinSal, 8,1)
AddBrokenLinearReclassificationRow(OxygenMinSal, 10,1)
```