REPORT N^{O} 70002867-001

SCAPE+ USER MANUAL

INTEGRATED COASTAL SEDIMENT SYSTEMS (ICOASST)

PUBLIC

OCTOBER 2016

SCAPE+ USER MANUAL

ICOASST

University of Southampton - NERC

Technical Manual (Revision 1)
Public

Project no: 70002867-001
Date: October 2016

WSP | Parsons Brinckerhoff
1st Floor Keble House
Southernhay Gardens
Exeter

Tel: +44 (0) 1392 267 500
Fax: +44 (0) 1392 267 599
www.wsp-pb.com

QUALITY MANAGEMENT

ISSUE/REVISION	FIRST ISSUE	REVISION 1	REVISION 2	REVISION 3
Remarks				
Date	30th July 2016	21st October 2016		
Prepared by	Mike Walkden & John Barnes	Mike Walkden		
Signature				
Checked by	Rachel Bird			
Signature				
Authorised by				
Signature				
Project number	70002867-001	70002867-001		
Report number	70002867-001	70002867-001		
File reference				

PRODUCTION TEAM

WSP | PARSONS BRINCKERHOFF

Project Director	Mike Walkden
Project Manager	Eddie Crews

SUBCONSULTANTS

FORTRAN Programmer and OpenMI specialist	John Barnes

TABLE OF CONTENTS

TABLES

FIGURES

APPENDICES

1 INTRODUCTION

1.1 BACKGROUND

This document has been prepared to describe the steps necessary to run a SCAPE+ model using the SCAPE executable (SCAPE.exe).

SCAPE (Soft Cliff And Platform Erosion) was developed to represent the mesoscale evolution of ‘cliffed’ coasts, and was first described by Walkden and Hall (2005). SCAPE+ is a version that has been produced within the Integrated Coastal Sediment Systems (iCOASST) project.

These developments have been made to make the model more accessible, easier to use and also to improve the representation of some processes. The stochastic cliff failure model RegFalls3 (developed by Jim Hall, and described in Dawson *et al*, 2009) has also been included.

SCAPE+ has been made ‘OpenMI compliant’ to support model coupling.

Readers of this document should refer to the Doxygen HTML files for further details of model parameters.

1.2 LICENSE

The SCAPE+ executable and the source code from which it was compiled are available under the GNU GENERAL PUBLIC LICENSE (Version 3, 29 June 2007, see Appendix A).

1.3 KEY CONCEPTS

The development of SCAPE was informed by a systems perspective of coastal geomorphology. In this view, the coast perpetually emerges from the interaction of a set of elements (in this case cliffs, shore platforms and beaches) in response to marine forcing. Order emerges from these interactions due to feedback pathways. Generally speaking negative feedback allows the emergence of dynamically stable forms.

A key reason for the existence of coastal models is to inform management decision making, and moreover, coastal management is often a key influence on coastal morphology. For these reasons SCAPE was designed from the outset to represent the effects of certain coastal management interventions.

A modelling tool with short simulation times allows multiple simulations, which is a key advantage when dealing with the uncertainties that are inherent in predictive modelling. To gain this advantage is not practical to develop a model (particularly one intended to represent ‘whole’ systems) from detailed process descriptions. For this reason, relatively abstract process representations were sought and (where necessary) developed. Some behavioural rules were also devised and used. The network of components and intersections within SCAPE are shown in Figure 1-1.

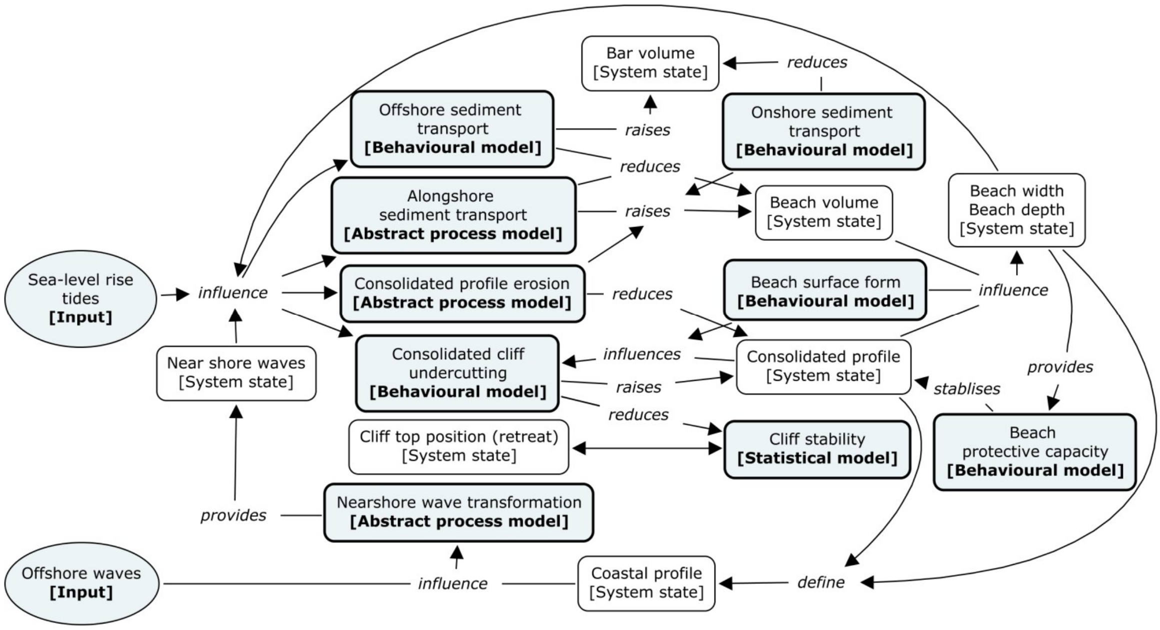


Figure 1-1. Components and interactions within SCAPE

Coastal form is represented within SCAPE with one or more shore profiles. These profiles are aligned perpendicular to a straight model baseline. A ‘quasi 3D’ representation of a coast can therefore be created with a series of shore profiles assembled along the same baseline.

A key strength of SCAPE is its representation of the shaping of a surface of consolidated material. These erode through time in response to wave attack and cannot accrete. Each consolidated profile is defined on a vertical grid, essentially as a set of distances from the baseline (one for each vertical position in the model, see Figure 1-2). These are represented by the model variable `cliffy`. SCAPE can also represent beach profiles, which are also defined on the vertical grid. The beach volume is calculated during model runtime, or is predefined. The position of the beach is then calculated by shifting the position of the beach profile horizontally, until the volume it encloses with the consolidated profile is correct. Its surface is assumed to follow a ‘Bruun’ form.

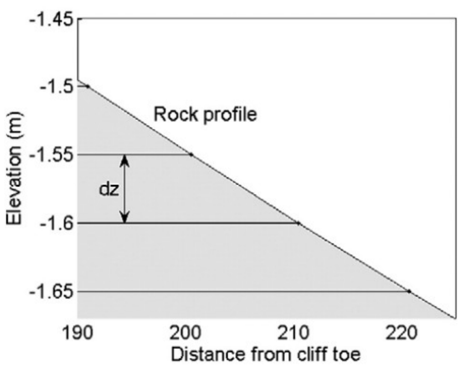


Figure 1-2. Local detail from a model showing the seaward face of a stack of SCAPE elements

It is crucial for a (Quasi 3D) model to represent alongshore interactions, particularly when the effects of a coastal management structure are being considered. This happens in SCAPE largely through the exchange of beach material between neighbouring profiles.

SCAPE is typically used to represent either (1) the very long periods of time over which consolidated shore profiles emerge, or (2) shorter periods of time over which they respond to changes to their input forcing, such as accelerated sea level rise or the construction of a seawall.

1.4 CONTROLLING SCAPE+

During initialisation SCAPE+ reads model parameters from a file named `setup.txt`, and also various characteristics of the shore from a set of shore state input files, as illustrated in Figure 1-3.

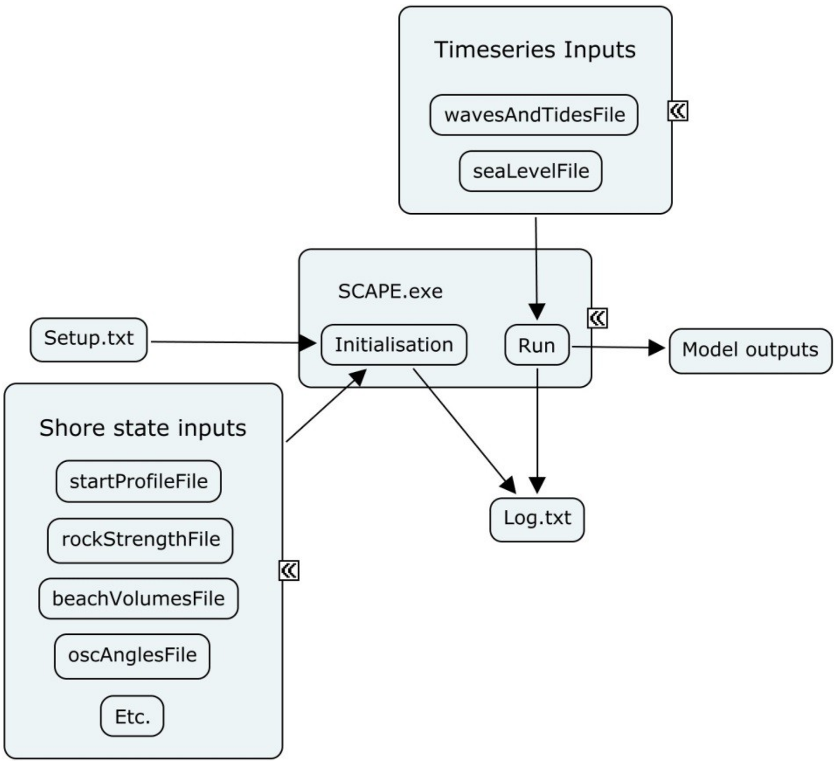


Figure 1-3. Running a SCAPE+ model

Whilst running, SCAPE+ reads inputs from timeseries files describing wave conditions and high water levels (every tide) and mean sea levels (every year). Model parameters are written to a log file, which is also used to record details of the simulation, including error messages. The inputs and SCAPE.exe file must be in the same directory.

Further details on the `setup.txt` file and the various input and output files are given below, in Sections 3 and 4.

1.5 MODEL HISTORY AND SELECTED PUBLICATIONS

The development of SCAPE began at Bristol University, in 1999. The first site to be modelled was the Naze peninsula in Essex, UK (see Walkden and Hall, 2005). Key strengths of that work included:

- good representation of shore profile emergence despite the use of quite abstract descriptions of the involve processes;
- a long term (50 year) prediction timeframe;
- the representation of the effect of structures (groynes, seawalls, and hard points);
- the explicit representation of sea level rise; and
- short run times and probabilistic application.

Following this success, SCAPE was used to represent the coast of North Norfolk (later published by Walkden and Hall, 2011) for a coastal strategy study on behalf of North Norfolk District Council. In order to capture the large spatial scale, elements of the code were simplified, including the representation of the effect of groynes. Key strengths of that study were:

- the representation of a long extent of coast (35 km);
- successfully reproducing the long historic record change of this coast (circa 100 years);
- Simulation of the local and alongshore effects of coast protection structures; and
- exploration of alternative scenarios of coastal management and future sea level rise.

The Norfolk model was subsequently developed with funding from the Tyndall Centre for Climate Change Research and adopted as a geomorphological engine in its Regional Coastal Simulator (see Dickson *et al*, 2007, Walkden, *et al*, 2015). Key strengths were:

- the representation of 50km of coast (Blakeney Spit to Winterton Ness);
- calibration/ validation against a 117 year historic record;
- direct coupling with simulations of flood risk in the Norfolk Broads (see Dawson et al, 2009);
- inclusion of rule-based beach nourishment; and
- probabilistic treatments of uncertainty.

In addition a stochastic model of coastal landsliding (named RegFalls3) was developed by Jim Hall, and designed to be suitable for SCAPE output. This is described in Dawson *et al*, 2009.

In the same period, SCAPE was used to model two sites as part of the Management of Cohesive Foreshores project, (Defra/ EA, 2007, Cooper *et al*, 2007) which was funded by the Environment Agency. These sites were Warden Point (on the Isle of Sheppy) and Easington (Holderness).

The Tyndall Centre and the Environment Agency also funded research that used SCAPE to explore the basic relationship between sea level rise and shore recession rate. Through a large set of 2D simulations, sensitivity to increased sea level rise was explored as certain parameters were varied (including tidal range, wave height and rock strength). A simple relationship was found, and this was expressed in a simple form, tailored for use in coastal management projects (Walkden and Dickson, 2008). That work was subsequently given a broader theoretical basis (Ashton *et al*, 2011).

In 2008 SCAPE was used within the Defra/ Environment Agency R & D project *Characterisation and prediction of large scale, long-term change of coastal geomorphological behaviours* in a proof of concept to explore the coupled behaviour of an (hypothetical) open coast and estuary; the latter being represented by an ASMITA model (Walkden and Rossington, 2009).

SCAPE has also been coupled to TELEMAC (Chini *et al*, 2010) and used to represent sites (inspired by locations on the Isle of Wight) with across-shore variations in geology (Carpenter *et al*, 2014).

In recent years most applications have been largely commercial, and have included new models of the shores of Drigg (Cumbria), West Somerset, and the Wash. SCAPE simulations have also been used to inform work on coastal catch-up in the Clacton area.

The Environment Agency project Cliff and Shore Sensitivity to Accelerated Sea Level Rise used the SCAPE model in various ways, including modelling (in 2D) of the shores of Holderness (unsuccessfully), Nash Point (Glamorgan), Birling Gap (Sussex), Happisburgh (Norfolk) and Drigg (Cumbria) (see Walkden et al, 2014). This study included a very large set of simulations that mapped the transient stage of shore response to changes in rates of sea level rise (i.e. before the emergence of dynamic equilibrium). The project also modelled coastal response to the removal of coastal structures, using observations from the Happisburgh coast (Walkden *et al*, 2015b).

Within the iCOASST project SCAPE+ has been used to simulate;

- very long term shore profile development;
- the coupled behaviour of a cliff and barrier/beach system, similar to the region of Suffolk between Southwold and Kessingland (UK) (Walkden *et al*, 2015c); and
- the open coast either side of the Deben estuary (Suffolk, UK) within a FluidEarth composition.

1.6 SCHEMATISATION

The reader should refer to the various publications described above for detailed information on the structure of SCAPE models. This section provides some additional information relevant to the definition of inputs and outputs of SCAPE+.

TIMESTEP

Every model timestep (one tidal period), data describing wave height, period and direction, tidal amplitude, and rate of sea level rise are read from input files and the system state (rock profile, beach width, beach depth and nearshore wave condition) is recalculated. Wave conditions are assumed to be constant throughout a tidal timestep, whilst water levels are assumed to vary sinusoidally.

SHORE PROFILE

Each consolidated shore is composed of a stack of horizontally aligned elements, the seaward faces of which form the exposed profile.

The vertical difference between the lowest and highest element is defined by the parameter `sectionHeight` (see Figure 1-4).

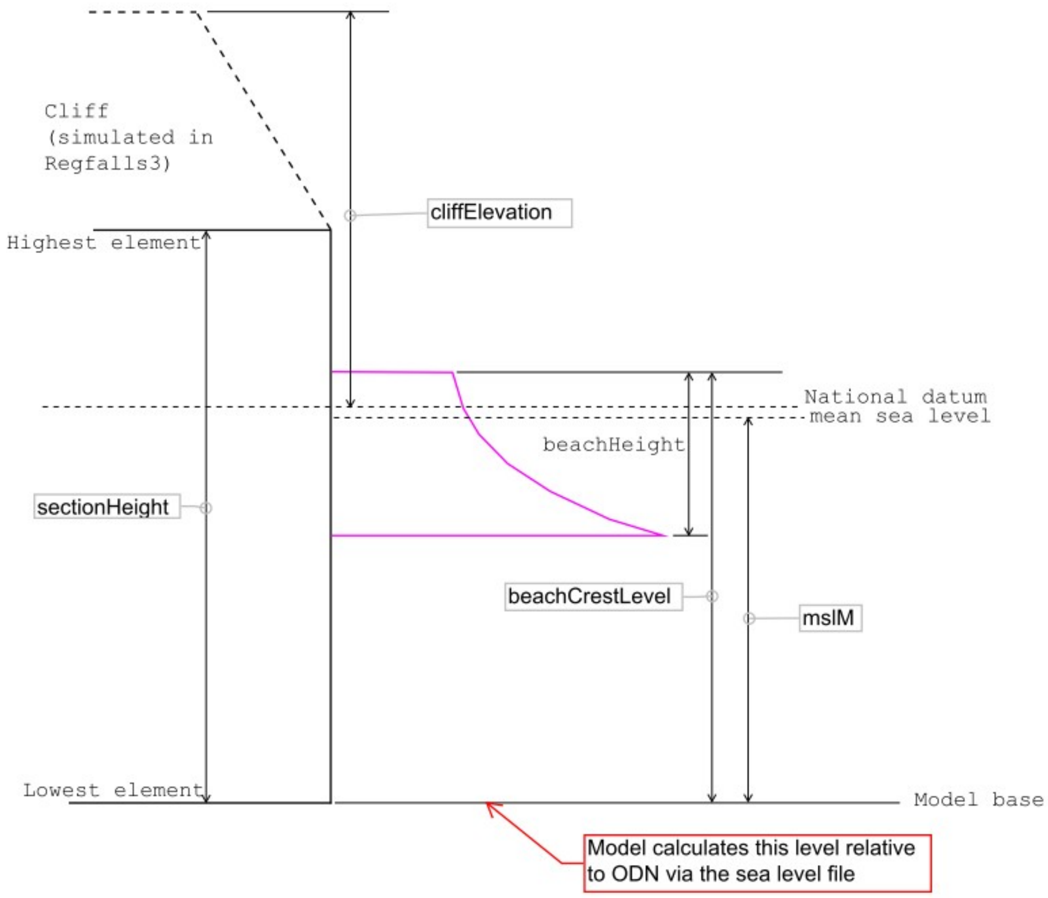


Figure 1-4. Conceptualisation of the shore profile

The vertical elevation of each element is defined (internally) relative to mean sea level via the parameter `mslM`. The user may wish to start the model such that the sea level is not precisely at the boundary between two SCAPE elements. This is allowed via the parameter mslOffset, as illustrated in Figure 1-5.

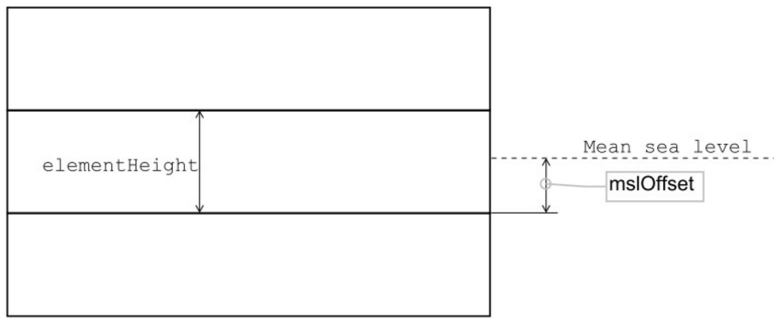


Figure 1-5. Refinement of the relationship between the model and sea level

The beach profile (which is assumed to follow a Bruun/ Dean curve, defined by the parameter `bruunConst`) is defined in a similar manner to the consolidated profile. Generally a beach is not needed across the whole vertical range of the consolidated profile, and so the vertical extent of the beach is defined by a specific parameter (`beachHeight`). The elevation of the upper limit of the Bruun profile is defined relative to the model base level by `beachCrestLevel`. The upper surface of the beach (i.e. landward of the top of the Bruun curve) is controlled via `bermSlope`.

It should be noted that no distinction is initially made between the cliff face and the shore profile; this boundary emerges through model iteration. The highest element is therefore typically set to be a small height above the expected cliff toe.

Cliff landsliding is represented in the following two ways:

1. (within SCAPE) the cliff face is assumed to fail periodically, following a predefined number of erosion events (controlled by `slumpPeriod`). The purpose of this internal calculation is to account for sediment released by cliff erosion.

2. (within RegFalls3) following the SCAPE model run, the predicted shoreline retreat can be passed to RegFalls3 to generate probabilistic projections of cliff top retreat.

It should be noted that SCAPE+ has been developed (within the iCOASST project) to include a *horizontal* grid of eroding elements, coupled with the original vertical grid. This was implemented to:

- allow greater flexibility in shore profile shape;
- allow across-shore variation in material strength;
- support the representation of a coastal barrier; and
- support (ultimately) the capability for backshore material to accrete and consolidate.

The parameter `useVAgrid` is used to activate the grid, which then requires the input files `baseGeologyFile.txt` and `layerDataFile.txt`.

Good progress was made in the implementation of this horizontal grid, however resource was not available to undergo comprehensive testing, and so this part of the model cannot yet be considered properly operational. Consequently this user manual does not describe the horizontal grid, although references are made to optional input parameters that relate to it.

BASELINE

Each profile is located at a regular interval (`sectionWidth`) along the baseline; these are referred to below as ‘Y’ sections (see Figure 1-6). Y sections are separated by ‘Q’ sections, at which alongshore sediment transport calculations are made.

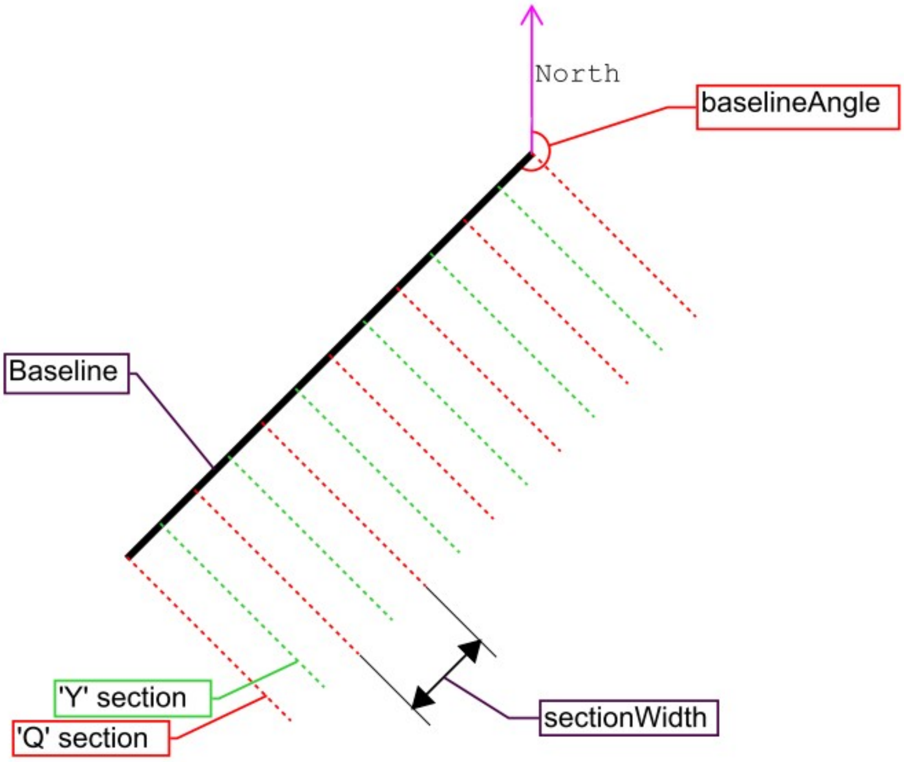


Figure 1-6. Arrangement of ‘Y’ and ‘Q’ sections

WAVES

Wave transformation due to shoaling and refraction is calculated using linear wave theory. Waves are assumed to transform across parallel offshore contours. The angle of the contour is defined via the input file `oscAnglesFile.txt`. One angle is required for each Q section, and is defined as illustrated in Figure 1-7.

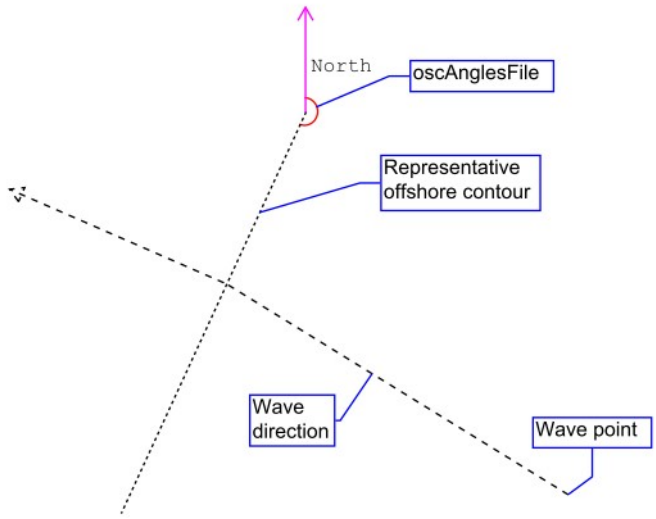


Figure 1-7. Definition of offshore contour angle

ALONGSHORE SEDIMENT TRANSPORT

SCAPE calculates alongshore beach transport processes in a manner similar to that used in 'one-line' modelling, and so useful insight can be gained by reviewing literature describing such modelling tools and techniques (see for example the US Army Corps of Engineers GENESIS manuals, http://chl.erdc.usace.army.mil/chl.aspx?p=s&a=Software;34).

The original SCAPE model calculated alongshore transport rates using the CERC equation (Hanson and Kraus, 1989). This requires the definition of the 'CERC constant' for each Q section, and these values are input via the input file `cercConstants.txt`.

In SCAPE+ an alternative equation developed for sand gravel and shingle has been added (van Rijn, 2014). This is activated if the model parameter `useAltSedimentTransport` is included in `setup.txt`, and requires the definition of the median grain size (d50) via the model parameter `medianGrainsize`.

1.7 CROSS-SHORE MOVEMENT

The SCAPE code was originally developed to model a site with a sparse beach at which cross-shore sediment motion could be neglected. Such an assumption is inappropriate at many locations, where cross-shore sediment movement during storms plays an important role in allowing erosion of the shore platform and cliff.

For this reason representation of cross-shore sediment movement was introduced to SCAPE when it was applied to the Norfolk coast (as described in Walkden and Hall, 2011).

No method was found in the literature that met the requirements of the approach for speed, simplicity of implementation, and capability of long-term simulation. A computing-intensive approach could not be justified since cross-shore processes were held to be of limited importance in governing long-term recession. Cross-shore processes were therefore treated in a parsimonious manner, based on parameterisation of a more sophisticated model, while adopting behavioural rules where necessary.

The sediment transport model COSMOS (COastal zone Simulation Model, Nairn and Southgate, 1993) was applied to the Norfolk study (by HR Wallingford) to identify the conditions under which beach material moves seaward from the beach and the rates of transport as functions of wave height and period. The COSMOS simulations did not account for wave reflection at structures or at the face of cliffs.

Under storm conditions the beach is assumed to divide into a steep upper and more gently sloping lower beach. In addition an offshore store is assumed to exist. Both beach surfaces are represented with Bruun curves; the upper having a Bruun (or Dean) constant 0.05 greater than the normal beach, and the lower 0.05 less. At the onset of a storm, the beach volume is divided equally between the two parts; then cross-beach and offshore sediment transport rates are identified from tables populated with the COSMOS output.

These are used to calculate sediment movement from the upper to the lower beach and from the lower beach to the offshore store. Once the storm conditions cease, the upper- and lower-beach volumes are recombined and the beach surface shape is returned to the normal Bruun curve. No material is truly lost offshore.

The rate at which material returns from the offshore store to the beach is unknown, and a suitable model has not been identified in the literature. Consequently a simple behavioural rule is used, based on an order of magnitude estimate of the onshore flux. It is assumed that material in the store returns a proportion of its total volume every tide (P_{BV}). Given that it is extremely difficult to estimate what this proportion is and how it varies, the fewest assumptions have been made, which are: (i) invariant P_{BV} and (ii) an order of magnitude–based selection of its value. Clearly 100% return of material per tide is too great; 10% also seems very large, and so P_{BV} = 1% is chosen.

In SCAPE+ the cross-shore sediment transport rates derived for the Norfolk study are provided as default (as described in Section 3), and the user is able to introduce alternative values.

1.8 CLIFF FACE (REGFALLS3)

RegFalls3 simulates the cyclic failure of a coastal cliff face, resulting from the progressive removal of material from the base. The timing of such landslides cannot be predicted precisely. However, knowledge of the rate of shoreline retreat (from SCAPE) can be combined with an assessment of the geotechnical characteristics of the slope to generate an approximate probability distribution of the possible cliff top location following failure (Hall *et al*. 2000).

The length of coast within a given Y section is assumed to behave in a uniform way. Within each section the cliff can be expected to fail when it reaches an average angle α_f and will, after failure, adopt an angle α_s. Even if all the required information were available neither α_f nor α_s can be predicted precisely because of uncertainties in our understanding of the processes of coastal landsliding. This uncertainty in α_f and α_s has been included in the RegFalls3 analysis by representing both values as Normally-distributed random variables, with means and variances obtained from a geomorphological assessment. The situation is illustrated in Figure 1-8.

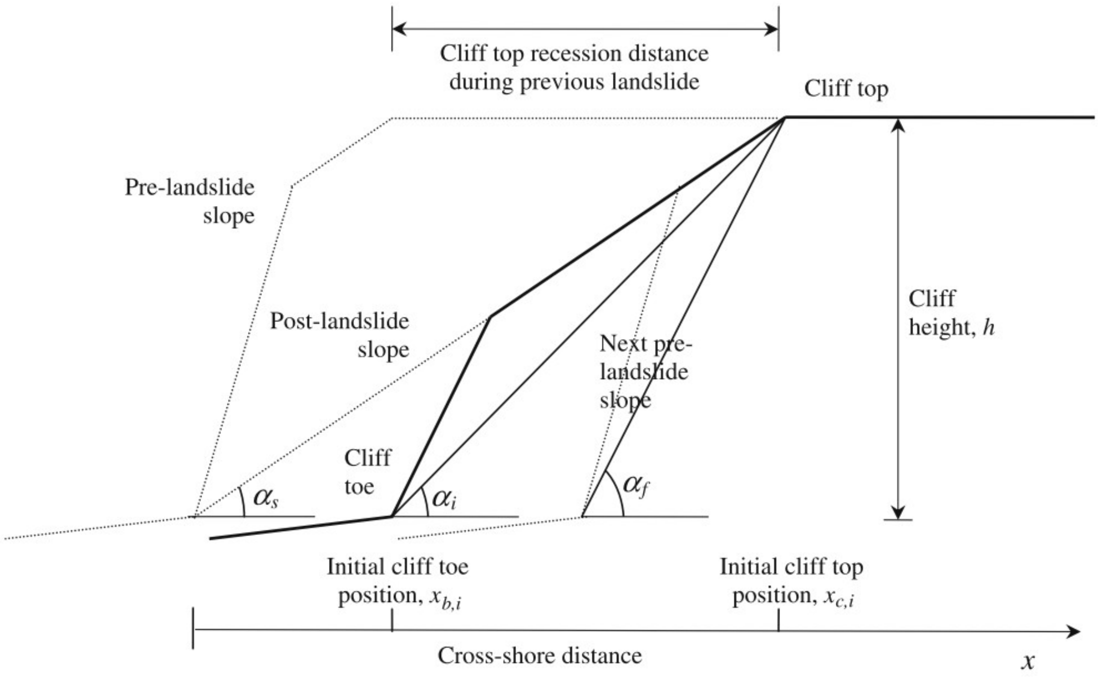


Figure 1-8. Representation of landsliding within RegFalls3

By default RegFalls3 is not run during a SCAPE+ simulation. It may be activated by the user via the `setup.txt` file, by setting the `nCliffSimulations` parameter to a value greater than zero. If this is done then the model will open `cliffData.txt`, which is required by RegFalls3 to run (and which the user must prepare).

1.9 COASTAL MANAGEMENT INTERVENTIONS

SCAPE+ allows representation of two types of coastal management intervention, seawalls and groynes.

SEAWALLS

A seawall is recognised in the model (by including `includeSeawall` in the model setup file) as a position (distance from the model baseline) behind which a profile may not erode. Seawalls may be controlled in either of the following two ways:

1. by defining the year at which it is constructed *and* the position (distance from the model baseline) at which it appears, or

2. by defining the year at which it is constructed *and* a reference level (relative to the model base) on the consolidated profile at which it will appear.

Seawall failure years may also be defined.

GROYNES

The effect of groynes may also be represented at chosen Q sections, through their effect on alongshore sediment transport. The user specifies groynes by defining both the year in which they are constructed and also a table of groyne effect parameters (see below).

1.10 BEACH LEVEL FOR SEAWALL STABILITY CALCULATIONS

The SCAPE code was originally developed to model a site with a continuous cliff, and where seawall stability could be neglected. Such an assumption is inappropriate at many locations, where seawalls protect low-lying hinterland, which may be vulnerable to coastal flooding.

In such settings information on the beach level at the structure is of potential value in determining seawall stability (and therefore inundation probability). For this reason the representation of the beach profile was extended, when SCAPE was applied to the Norfolk coast.

The method that was chosen as appropriately relevant to the setting and undemanding of processor time was the beach profile model proposed by Vellinga (1986); this is illustrated in Figure 1-9.

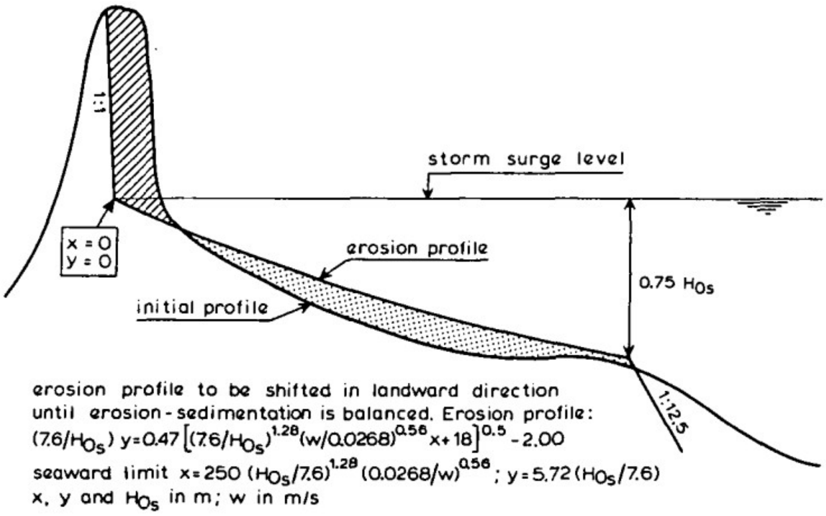


Figure 1-9. Beach profile model provided by Vellinga (1986)

By default the code that calculates ‘Vellinga’ beach profiles is not used. However the user can activate it via the `setup.txt` file using the `vellingaActive` keyword (see below).

If activated then a separate .txt file is read (`vellingaSections.txt`) that lists the sections that the code is to be applied to. It may be noted that the code will only operate on sections where (and when) a seawall is present.

1.11 OPENMI

SCAPE+ complies with the OpenMI standard. This means that SCAPE+ models may be set up to exchange information with other (OpenMI compliant) models whilst they run simultaneously. FluidEarth (formerly known as OpenWEB) is an HR Wallingford initiative to promote integrated modelling. FluidEarth software may be used to build compositions of OpenMI compliant models, through its interface software (Pipistrelle). Instructions on how to build SCAPE+ models into OpenMI compositions is given in Appendix B.

2 RUNNING A BASIC SCAPE+ MODEL

This section is intended to act as a guide for the construction of a simple and generic SCAPE+ model. The characteristics of this model are:

- 10 model sections, each spaced 500 m apart;
- A total vertical extent of 20m, composed of (400) elements, each 50mm high;
- Simulation for 1000 years, from 1015 AD to 2015 AD;
- A small beach; and
- Sea level rise of 2 mm/y.

It is expected that the reader will wish to refer to subsequent sections for further information on the various input parameters and input/ output files.

2.1 SETUP.TXT

In order to run, the SCAPE+ executable must be accompanied by a `setup.txt` file, containing (at least) the following parameters:

```
STARTYEAR
#
ENDYEAR
#
BEACHCRESTLEVEL
#
BASELINEANGLE
#
DEPTHOSCMSL
#
```

The following values have been chosen in this example:

```
STARTYEAR
1015
ENDYEAR
2015
BEACHCRESTLEVEL
17.4
BASELINEANGLE
220
DEPTHOSCMSL
7
BRUUNCONST
0.2
```

In other words, the simulation will begin in year 1015AD and end in the year 2015AD; the upper surface of the beach (or more precisely the highest elevation in the curving face of the beach profile) will be 17 metres above the base of model; the baseline is oriented at 220 degrees, and the elevation difference between mean sea level and the offshore contour is 5 metres. The Bruun constant, which defines the steepness of the beach face, is set to 0.2.

The `setup.txt` file can be created simply by copying the list above into an otherwise empty text file named ‘setup.txt’.

More lines will be added in this example including:

```
STARTMODE
NSECTIONS
10
SECTIONWIDTH
500
SECTIONHEIGHT
20
MSLM
15
```

The first line tells SCAPE+ to operate in Start Mode (as opposed to Restart Mode, see below). The remainder of these additional lines are redundant, because the values chosen are equal to the default settings. They have been added here to highlight the fact that this model will represent: a 4.5 km length of coast, composed of 10 sections spaced at 500 m intervals, each of which is 20 metres high. In addition the mean sea level at the start of the simulation is 15 m above the base of the model.

The executable must also be accompanied by the following set of input files:

```
profOffsets.txt
cliffElevations.txt
beachContent.txt
oscAngles.txt
cercConstants.txt
rockStrength.txt
seaLevel.txt
wavesAndTides.txt
shapeFunctionDrift.txt
shapeFunctionErosion.txt
clut.bin
cglut.bin
```

Each of these is explained below.

PROFOFFSETS.TXT

This file provides values for the horizontal offset of the top consolidated profiles from the baseline. A value is required for each model section. In this example a value of 5000 m will be used for each, and so `PROFOFFSETS.TXT` looks like this:

```
5000
```

```
5000
5000
5000
5000
5000
5000
5000
5000
5000
```

CLIFFELEVATIONS.TXT

This file provides values (in m above OD) for the level of the cliff top, for each model section. In this example we will assume that the cliff is 12 m high throughout, and so the contents of the file will be:

```
12
12
12
12
12
12
12
12
12
12
```

BEACHCONTENT.TXT

This file provides values for the proportion (from 0 to 1) of the rock that will form beach-suitable material after it has been eroded from the platform and cliff. A value is required for each model section. In this case it is assumed that 75% of the rock volume will be lost (carried away as fine sediment) and 10% will pass into the beach. Consequently this file will contain the following values:

```
0.1
0.1
0.1
0.1
0.1
0.1
0.1
0.1
0.1
0.1
```

OSCANGLES.TXT

As noted above, this file provides values for the angle of a representative offshore contour. The contents must be a value (in degrees) for each Q section (note that the number of Q sections is equal to the number of Y sections plus 1). In this example the offshore contour is assumed to be straight, and to be at 200 degrees relative to north.

CERCCONSTANTS.TXT

This file provides values for CERC coefficients, which are constants used in the (default) alongshore sediment transport equation. The file must contain a value for each Q section (note that the number of Q sections is equal to the number of Y sections plus 1). In this example the CERC constant is assumed to be 0.77.

ROCKSTRENGTH.TXT

This file provides values for the strength of the platform/ lower cliff material. It contains, for each cliff element a value for each model section. In this example a value of $1x10^6$ $m^{9/4}s^{2/3}$ is chosen for the rock strength, and so the `rockStrength.txt` file is simply composed of a large matrix of the value `1000000`, in this case 10 columns wide, and 400 rows deep.

SEALEVEL.TXT

This file must contain two columns of data, the first representing year (in the Julian calendar) and the second representing the sea level in that year (in m relative to the national datum). The data in this file must, of course, contain the period bounded by `STARTYEAR` and `ENDYEAR`.

In this example we assume that the rate of sea level rise is 2 mm/y throughout the simulation, with an initial value of -2 m. The algorithms dealing with sea level rise require a value for the year after the end of the simulation, and therefore the start and end of the file `seaLevel.txt` is as follows:

```
1015     -2
1016     -1.998
1017     -1.996
1018     -1.994
1019     -1.992
1020     -1.99
…        …
2010     -0.01
2011     -0.008
2012     -0.006
2013     -0.004
2014     -0.002
2015     0
2016     0.002
```

WAVESANDTIDES.TXT

This file provides values for tidal amplitude (in m relative to mean sea level), and wave period (seconds), height (metres) and angle (degrees, representing the direction the wave is travelling from). One row of conditions is needed per timestep. SCAPE simulations normally represent periods of time that are longer than the period for which wave data is available. If the wave record is shorter than the simulation period then the code will read to the end of the `wavesAndTides.txt` file and then start reading again at the beginning.

A `wavesAndTides.txt` file is provided with this example, the first ten lines are as follows:

```
1.84   8.3   0.48   90
2      8.6   0.62   91
1.74   7.8   0.73   89
1.57   7.5   0.43   88
0.96   3.7   0.36   170
0.9    6.2   0.88   135
0.56   5.4   0.6    145
2.06   3.9   0.35   171
1.27   4     0.33   173
1.72   4.2   0.37   171
```

SHAPEFUNCTIONDRIFT.TXT AND SHAPEFUNCTIONEROSION.TXT

These files contain shape functions describing cross-shore distributions of erosion and alongshore sediment transport. Versions are available with the SCAPE executable.

CLUT.BIN AND CGLUT.BIN

These files contain look-up tables for wave celerity and wave group celerity. Versions are available with the SCAPE executable.

OPTIONAL PROFILE OUTPUT PARAMETERS

The set of inputs described above are sufficient for SCAPE+ to run. However, they will not cause the model to output shore profile data, which are normally useful to have. To output these profiles add the following to the setup file:

```
FIRSTYEARPROFILEOUTPUT
1015
LASTYEARPROFILEOUTPUT
2015
FIRSTSECTIONPROFILEOUTPUT
1
LASTSECTIONPROFILEOUTPUT
10
PROFILEOUTPUTTIMESTEP
100
```

This will tell SCAPE+ to output the profile of section numbers 1 to 10 (i.e. all of the model sections) from year 1000 to year 2000, in time steps of 100 years.

BOUNDARY CONDITIONS

In this simple example, the alongshore boundaries will be set to prevent any influx of beach material. This will ensure that the emerging beach will only contain locally released sediment. This is achieved by adding the following lines:

```
QPMAXBOUNDARYRIGHTIN
0
```

As may be guessed, these keywords represent the maximum potential alongshore transport (in m^3/tide) allowed at the right and left boundaries respectively.

DYNAMIC LINK LIBRARIES

Depending on the compiler used to create the SCAPE executable file, some dynamic link library (`#.dll`) files may be required for SCAPE to run. A set are included with the downloadable SCAPE executable, to support its operation. These should be placed in the same folder as the SCAPE executable.

RUN

The SCAPE+ executable file will now run. A series of messages will appear, noting stages of the model initialisation. These will be followed by a simple count of the years being simulated, and then a further note on the total duration of the simulation. At the time of writing, this 1000 year simulation took approximately 2 minutes, to run.

An example of the screen output of this simple model is shown below

```
*** SCAPE Version 1.21

 Timeframe:
 From year  1015 output every     1 years
 Until year  2015
 Run duration:  706706 tides

 Assigning wave point numbers
 Structure control files loaded
 look-up tables loaded
C:\workshop\28_UserManualExample Year:  1015
C:\workshop\28_UserManualExample Year:  1016
C:\workshop\28_UserManualExample Year:  1017
C:\workshop\28_UserManualExample Year:  1018
C:\workshop\28_UserManualExample Year:  1019
C:\workshop\28_UserManualExample Year:  1020
…
…
C:\workshop\28_UserManualExample Year:  2009
C:\workshop\28_UserManualExample Year:  2010
C:\workshop\28_UserManualExample Year:  2011
C:\workshop\28_UserManualExample Year:  2012
C:\workshop\28_UserManualExample Year:  2013
C:\workshop\28_UserManualExample Year:  2014
C:\workshop\28_UserManualExample Year:  2015
```

```
*** Program finishing
run duration          57   -seconds          1   -minutes
0   -hours          0   -days
CPU time is    1.94637251      minutes
Final mslOffset:    1.9971671139613644E-003
SCAPE+  Finishing at year:  2015
```

OUTPUTS

Running the example model will generate the following series of output files:

```
log.txt
GROYNE_FACTORS.txt
CONTOUR_ROCK.txt
CONTOUR_SHORE.txt
LIBERATED_SED_BEACH.txt
LIBERATED_SED_FINES.txt
SEAWALL_PRESENCE.txt
SED_FLUX_L_BOUND.txt
SED_FLUX_R_BOUND.txt
SED_FLUX_POT_L_BOUND.txt
SED_FLUX_POT_R_BOUND.txt
SED_TRANS_ANN.txt
SED_TRANS_POT_ANN.txt
VOLUME_B_ANN_AVE.txt
```

The optional profile output parameters described above will cause the following files to also be produced:

```
ROCK_PROFILES.txt
BEACH_PROFILES.txt
```

Figure 2-1 shows data taken from the `ROCK_PROFILES.txt` and `BEACH_PROFILES.txt` files. It shows the final recorded shore profile at model section 5.

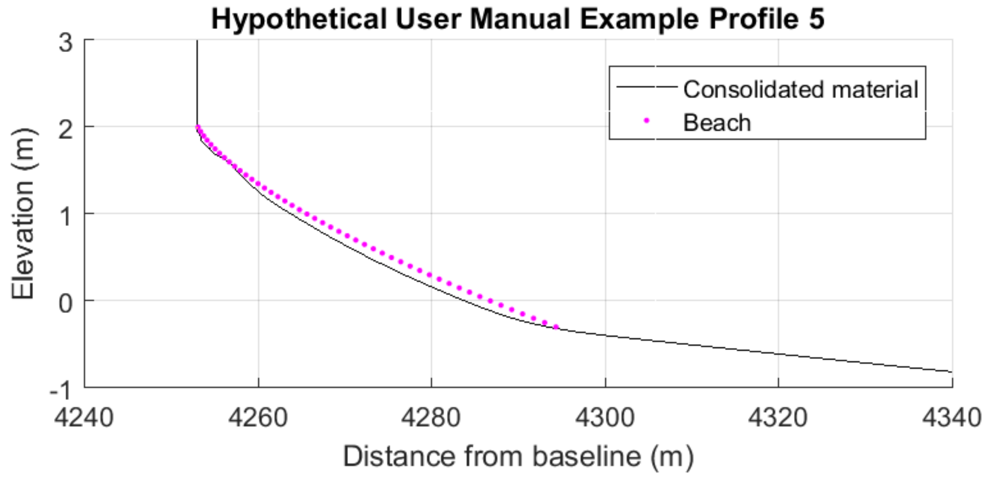


Figure 2-1. SCAPE+ output, profile 5 at the end of the example simulation

It can be seen that the beach extends for a little over 40 metres across the shore profile, between elevations of, approximately, -0.3m and +2m.

The plan shape recession may be seen by visualising data from `CONTOUR_SHORE.txt`. In Figure 2-2 the shore position (as recorded by `CONTOUR_SHORE.txt`) is shown every 100 years. The shore position begins as a straight line, 5000 m from the baseline (the distance set in `profOffsets.txt`). Around 200 metres of recession occurs over the next 100 years. By the end of simulation the overall recession of the shore approaches 1km for the lowest section numbers, and around 850m at the highest section numbers.

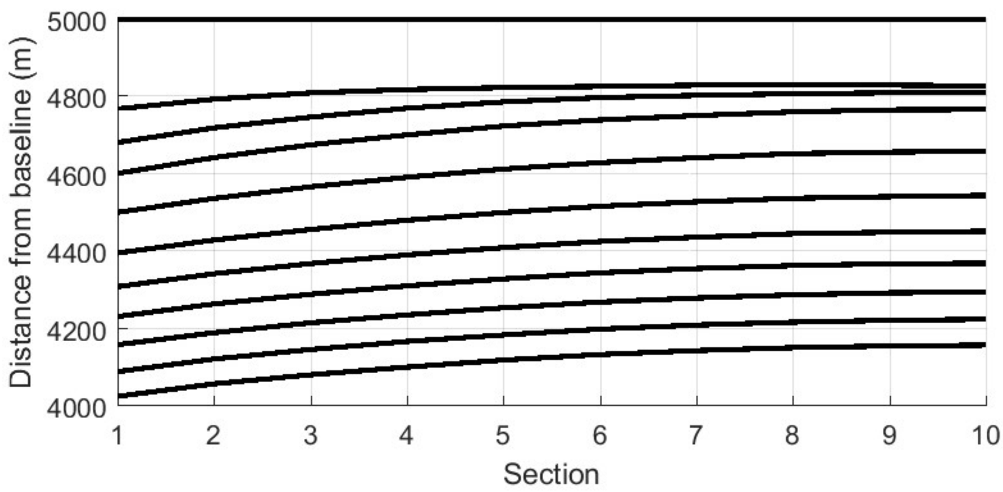


Figure 2-2. Shoreline planshape every 100 years, derived from CONTOUR_SHORE.txt

Information on the alongshore sediment transport rates can be seen in Figure 2-3.

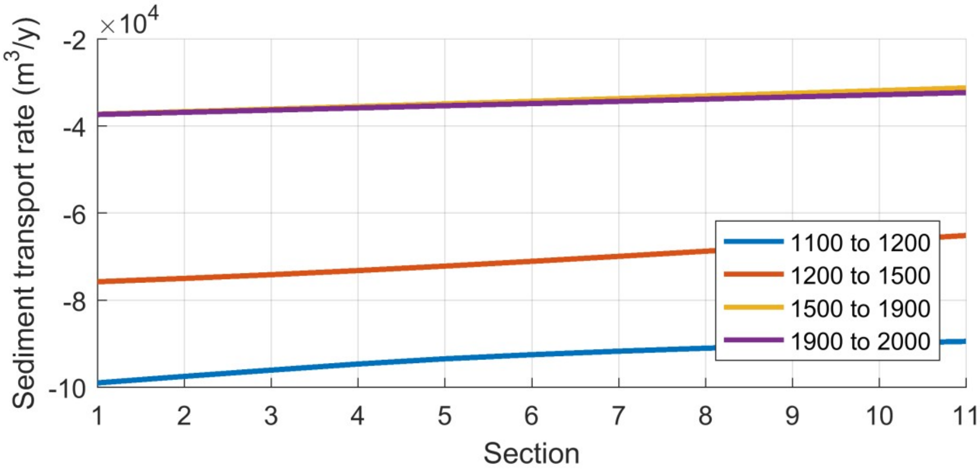


Figure 2-3. Alongshore sediment transport rates across four epochs, derived from SED_TRANS_ANN.txt

These transport rates have been calculated from `SED_TRANS_ANN.txt` and show average transport rates across four epochs (which are descried in the legend). The values are all negative (indicating movement towards lower section numbers). They range from 95,000 m^3/ year early in the simulation when recession (and therefore sediment release) is high down to less than 40,000 m^3/year at the end of the simulation. It may be noted that there is little change in alongshore sediment transport rate after the year 1500; by this stage the beach volume, shore recession and alongshore transport appear to have reached a state of dynamic equilibrium.

SCAPE+ outputs are described in more depth in Section 4, whilst Section 3 gives a more comprehensive description of SCAPE+ inputs.

3 MODEL INPUTS

3.1 RUNMODE

As noted above, SCAPE is typically used to represent either:

- the very long periods of time over which consolidated shore profiles take to emerge; or
- shorter periods over which they respond to changes in their forcing (such as an increase in sea level rise or the construction of a seawall).

Typically conditions at the start of a shorter simulation will be based on conditions at the end of a long simulation. For this reason SCAPE+ may be run under a ‘restart’ mode, in which a larger set of initial conditions are read in from files.

3.2 PARAMETERS

SCAPE reads model parameters, including any non-default names for other input files, from the file ‘setup.txt’ in the current working directory (see Table 1 and Table 2). Most of the model parameters can be defaulted. The parameters can be in any order in the `setup.txt` file. Reference should be made to the source module `setup.f95` and its documentation for a description of each of these parameters and their default values. The following sections give an overview of parameter usage.

SIMPLE KEYWORD/VALUE PARAMETERS

For these parameters, the parameter variable name in `setup.f95` and the keyword to be used in `setup.txt` to introduce a value for the parameter are the same (ignoring case).

In `setup.txt`, the parameter name is given at the start of a line and the parameter value appears on the next line.

Table 1. Simple keyword/ value parameters

Parameter Name	Default Value	Required?
`tidesPerYear`	`706`	
`startYear`		Yes
`endYear`		Yes
`spinUpToYear`	`startYear value`	
`beachZeroDelay`	`0`	
`lowResOutTimestep`	`200`	
`highResOutTimestep`	`1`	
`highResOutStartYear`	`startYear value`	
`firstYearTransportOutput`	`startYear value`	
`profileOutputTimestep`	`1`	
`firstYearProfileOutput`	No profile output	
`lastYearProfileOutput`	`endYear value`	
`firstSectionProfileOutput`	`1`	
`lastSectionProfileOutput`	`nSections value`	
`nSections`	`10`	
`sectionWidth`	`500.0D0`	
`sectionHeight`	`20.0`	

Parameter Name	Default Value	Required?
elementHeight	0.05D0	
elementWidth	1.0D0	
YmaxH		If useVAgrid is .true.
nLayers	10	
baselineAngle		Yes
firstActiveSection	1	
lastActiveSection	nSections value	
waveTideFileStart	1	
seawallBaseLevel		If includeSeawall is .true. and seawallMode is defineSeawallsByYear
mslM	15.0D0	
mslOffset	0.0D0	
depthOSCMsl	7.0D0	Yes
outputContourLevel	Level of the top element of the cliff	
minStormWaveHeight	2.0D0	
minHs	0.2D0	
maxPlatformSlope	70.0D0	
bermSlope	100.0D0	
initialBermStep	1.0D0	
minBermStep	1.0D-6	
maxBermStep	1.0D+6	
beachdisturbanceRatio	0.2D0	
beachReturnRatio	0.01D0	
talusStrengthRatio	0.1D0	
talusSlope	45.0D0	
maxBlockSize	2.0D0	
beachHeight	10.0D0	
bruunConst	0.2D0	
beachCrestLevel		Yes
minBoundaryTransportRatio	0.1D0	
qpMaxBoundaryRightIn	1.0D+9	
qpMaxBoundaryLeftIn	1.0D+9	
qpMaxBoundaryRightOut	1.0D+9	
qpMaxBoundaryLeftOut	1.0D+9	
excludeWaves	0	
switchTransportWhDiff	0	
nCliffSimulations	0	
celerityTablesAsTxtFiles	.false.	
slumpPeriod	10	
tidalAmpSection	1	
tidalAmpScaleLeft	1.0D0	
tidalAmpScaleRight	1.0D0	
medianGrainsize	6.0D-3	

KEYWORD-ONLY PARAMETERS

None of these parameters are required, since they all have default values.

The indicated keywords can be used at the start of a line in `setup.txt` to change the value from the default.

Table 2. Keyword only parameters

Parameter	Default Value	Keywords
runMode	restart	restartMode startMode
useVAgrid	.false.	useVAgrid
tidalRangeVariation	.true.	noTidalRangeVariation
includeSeawall	.false.	includeSeawall
seawallMode	defineSeawallsByYe ar	defineSeawallsByYear defineSeawallsByYearAndPosit ion
includeGroynes	.false.	includeGroynes
useAltSedimentTranspo rt	.false.	useAltSedimentTransport
vellingaActive	.false.	vellingaActive

WAVEPOINT PARAMETERS

The `nWavepoints` parameter is optional. If absent, one wavepoint is assumed, with default `fswp` and `depthwpmsl` values. Search for `nWavepoints` in `setup.f95` to find the default values.

If `nWavepoints` is present, it must be followed by the number of wavepoints, then the corresponding number of `fswp` and `depthwpmsl` values. Wavepoint 1 is associated with low-numbered sections in the model.

This is the input for the current default wavepoint parameters:

```
NWAVEPOINTS
            1
            1
     10.000000
```

This is an example taken from a model with two wave points:

```
nWavePoints
            2
            1            20
            7.17D0       7.07D0
```

WAVE EXCLUSION ANGLES PARAMETERS

The `WAVEEXCLUSIONANGLES` keyword is optional. If absent, `waveExclusionAngleL` and `waveExclusionAngleR` are defaulted. Search for these variables in `setup.f95` to find the default values (which are also shown below).

If the `WAVEEXCLUSIONANGLES` keyword is present, it must be followed by the values for `waveExclusionAngleL` and `waveExclusionAngleR`.

This is the input for the current default wave exclusion angles:

```
WAVEEXCLUSIONANGLES

      22.500000          22.500000
```

CELERITY TABLES PARAMETER

The `CELERITYTABLESASTXTFILES` keyword is optional, and relates to the two wave celerity look-up table files required for SCAPE+ to run (`clut` and `cglut`). If absent, the celerity tables have 5000 depth values and 221 period values and are read from binary files (`clut.bin` and `cglut.bin`).

If the `CELERITYTABLESASTXTFILES` keyword is present, it must be followed by values for the celerity table dimensions `periodValues` and `depthValues`. The celerity tables can also be provided in text form (`clut.txt` and `cglut.txt`).

This parameter allows the user to prepare and use celerity look-up tables that are larger than those provided by default.

CROSS-BEACH SEDIMENT TRANSPORT RATE PARAMETERS

The `CROSSBEACH_STR` keyword is optional. If absent, a default cross-beach sediment transport rate table is utilised. Search for `CROSSBEACH_STR` in `setup.f95` to find the default table.

If the `CROSSBEACH_STR` keyword is present, it must be followed by the number of wave heights and wave periods in the table, then by the wave height values, then for each height the wave period values and transport rate values.

This is the input for the current default cross-beach sediment transport rate table:

```
CROSSBEACH_STR
           4           3
       3.0000000          4.0000000          5.0000000          -1.0000000
       5.0000000          6.0000000         -1.0000000
     -0.55000000        -0.74000000        -0.66000000
       6.0000000         -1.0000000          0.0000000
      -1.0000000         -1.2700000          0.0000000
       6.0000000         -1.0000000          0.0000000
      -1.0600000         -1.4600000          0.0000000
       6.0000000         -1.0000000          0.0000000
      -1.0900000         -1.5500000          0.0000000
```

For each of the height values (4 in the example), there is a row of period values starting on a new line, followed by a row of transport values starting on a new line.

The number of period and transport values in each data row is set by the number of period bins specified in the first line of the table data (3 in the example). The period bins actually used for each height value can be less than the number of period bins specified in the first line of the table (the period value -1.0 indicates the last period bin for the current height). In the example there are three period bins for the first height bin, but only two periods bins for the other height bins.

The table above has been annotated in Figure 3-1 to further explain its contents.

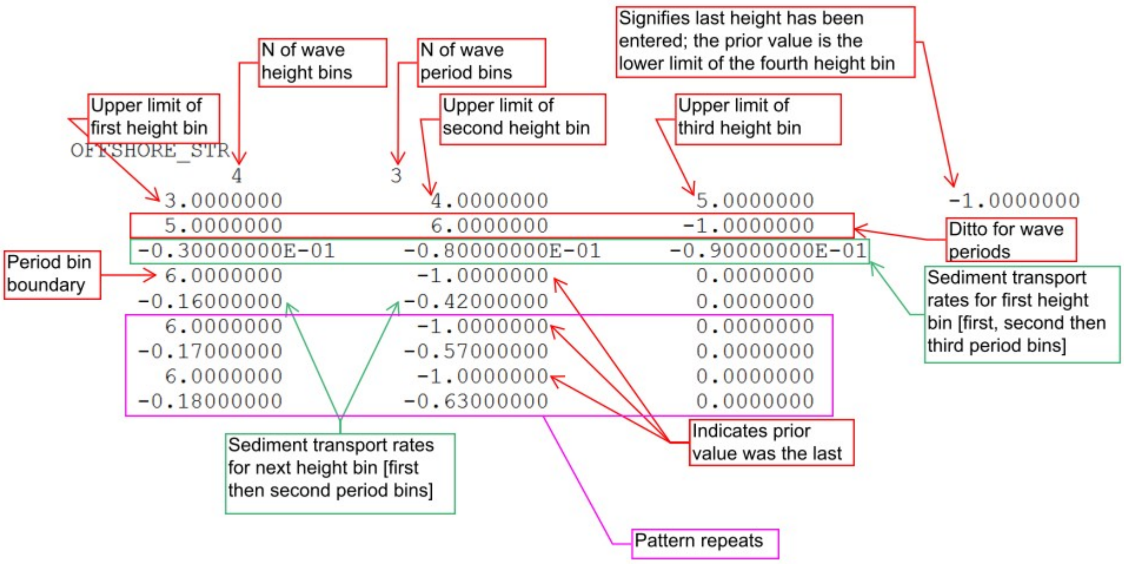


Figure 3-1. Annotated sediment transport table

This default describes the following sediment transport rates (cubic metres per metre of coast, per hour) under the wave height (*H*) and period (*T*) conditions shown.

H < 3		3 ≤ H < 4		4 ≤ H < 5		5 ≤ H	
T < 5	-0.55 m^2/h	T < 6	-1.0 m^2/h	T < 6	-1.06 m^2/h	T < 6	-1.09 m^2/h
5 ≤ T < 6	-0.74 m^2/h	6 ≤ T	-1.27 m^2/h	6 ≤ T	-1.46 m^2/h	6 ≤ T	-1.55 m^2/h
6 ≤ T	-0.66 m^2/h						

OFFSHORE SEDIMENT TRANSPORT RATE PARAMETERS

The `OFFSHORE_STR` keyword is optional. If absent, a default offshore sediment transport rate table is provided. Search for `OFFSHORE_STR` in `setup.f95` to find the default table (which is also shown below).

If the `OFFSHORE_STR` keyword is present, it must be followed by the number of wave heights and wave periods in the table, then by the wave height values, then for each height the wave period values and transport rate values.

This is the input for the current default offshore sediment transport rate table:

```
OFFSHORE_STR
            4           3
       3.0000000       4.0000000       5.0000000      -1.0000000
       5.0000000       6.0000000      -1.0000000
     -0.30000000E-01 -0.80000000E-01 -0.90000000E-01
       6.0000000      -1.0000000       0.0000000
     -0.16000000     -0.42000000       0.0000000
       6.0000000      -1.0000000       0.0000000
     -0.17000000     -0.57000000       0.0000000
       6.0000000      -1.0000000       0.0000000
     -0.18000000     -0.63000000       0.0000000
```

The structure of this table is identical to that used for the cross-beach sediment transport rate table described above. It describes the following default sediment transport rates (cubic metres per metre of coast, per hour).

H < 3		3 ≤ H < 4		4 ≤ H < 5		5 ≤ H	
T < 5	-0.03 m^2/h	T < 6	-0.16 m^2/h	T < 6	-0.17 m^2/h	T < 6	-0.18 m^2/h
5 ≤ T < 6	-0.08 m^2/h	6 ≤ T	-0.42 m^2/h	6 ≤ T	-0.57 m^2/h	6 ≤ T	-0.63 m^2/h
6 ≤ T	-0.09 m^2/h						

GROYNE EFFECT PARAMETERS

The `GROYNEEFFECT` keyword is optional. If absent, a default groyne effect table is provided. Search for `GROYNEEFFECT` in `setup.f95` to find the default table (which is also shown below).

If the `GROYNEEFFECT` keyword is present, it must be followed by the number of beach width values in the table, then by the beach width values, then by the groyne effect values. Values are within the range zero (no effect) and 1 (groyne completely blocks drift).

This is the input for the current default groyne effect table:

```
GROYNEEFFECT
            6
       50.000000          100.00000          150.00000          200.00000
       250.00000         -1.0000000
      0.95000000      0.90000000         0.50000000         0.40000000
      0.30000000         0.20000000
```

WAVE FACTOR PARAMETERS

The `MODIFYWAVEHEIGHT` and `MODIFYWAVEPERIOD` keywords are optional. If absent, wave height and period values are used as read from the `wavesAndTides.txt` file.

If either of these keywords is present, it allows lists of year and multiplying factor values to be supplied and must be followed by a count of the number of pairs of values, then by year values starting on a new line, then by the factor values also starting on a new line.

Here is an example where the wave height read from `wavesAndTides.txt` is left unchanged from the start of the run until 1989, then factored by 1.05 from 1990 to 2050, then factored by 1.1 from 2051 to the end of the run:

```
modifyWaveHeight

2

1990        2051

1.05        1.1
```

The year values must be increasing and the factor values must not be negative.

WAVE DIRECTION PARAMETER

The `MODIFYWAVEDIRECTION` keyword is optional. If absent, wave direction values are used as read from the `wavesAndTides.txt` file.

If the `MODIFYWAVEDIRECTION` keyword is present, it allows lists of year and angle change values to be supplied and must be followed by a count of the number of pairs of values, then by year values starting on a new line, then by the angle change values (in degrees) also starting on a new line.

Here is an example where the wave direction read from `wavesAndTidesFile` is left unchanged from the start of the run until 1989, then reduced by 10 degrees from 1990 to 2050, then increased by 10 degrees from 2051 to the end of the run:

```
modifyWaveDirection
2
1990        2051
-10         10
```

The year values must be increasing.

INPUT FILE NAME PARAMETERS

All input file name parameters are optional, having the default values shown in Table 3.

Whether a given input file is required during the run is indicated in the File Usage column.

Table 3. Input file name parameters

Name	Default Value	File Usage
oscAnglesFile	'oscAngles.txt'	Always
cercConstantsFile	'cercConstants.txt'	If useAltSedimentTransport is .false.
cliffElevationsFile	'cliffElevations.txt'	Always
wavesAndTidesFile	'wavesAndTides.txt'	
seaLevelFile	'seaLevel.txt'	
bermWidthsFile	'bermWidths.txt'	If runMode is restart
beachVolumesFile	'beachVolumes.txt'	
talusVolumeFile	'talusVolume.txt'	
offshoreBarVolFile	'offshoreBarVol.txt'	
rockStrengthFile	'rockStrength.txt'	If useVAgrid is .false.
beachContentFile	'beachContent.txt'	
startProfilesFile	'startProfiles.txt'	If runMode is restart and useVAgrid is .false.
profOffsetsFile	'profOffsets.txt'	If useVAgrid is .false.
baseGeologyFile	'baseGeology.txt'	If useVAgrid is .true.
layerDataFile	'layerData.txt'	
cliffDataFile	'cliffData.txt'	If nCliffSimulations is greater than 0
seawallConstructionYearsFile	'seawallConstructionYears.txt'	If includeSeawall is .true.
seawallRemovalYearsFile	'seawallRemovalYears.txt'	
groyneConstructionYearsFile	'groyneConstructionYears.txt'	If includeGroynes is .true.
groyneRemovalYearsFile	'groyneRemovalYears.txt'	
waveExclusionAnglesFile	'waveExclusionAngles.txt'	If excludeWaves is 2
startProfileFile	'startProfile.txt'	Not used
vellingaSectionsFile	‘vellingaSections.txt'	If vellingaActive

		`is .true.`

3.3 INPUT DATA FILES

The following sections describe the layout of data values in the input files. As far as possible values are read in free format but some values must appear as the first item on a line.

Looking offshore, Q point and shore section indices run right to left (i.e. index 1 is at the right-hand end of the model).

Whether a file is used in a run depends on a number of model parameters. See the table in the preceding section for details.

OSCANGLESFILE

Provides values for the angle of the offshore contour.

Contents: A value (degrees) for each shore section (1 : `nSections + 1`).

CERCCONSTANTSFILE

Provides values for CERC coefficients for the main (CERC) sediment transport equation.

Contents: A value for each Q point (1 : `nSections` + 1).

CLIFFELEVATIONSFILE

Provides values for the level of the cliff top.

Contents: A value (m from OD) for each shore section (1 : nSections).

WAVESANDTIDESFILE

Provides values for tidal amplitude, wave period, wave height and wave angle.

The first set of values read from the file is at line `waveTideFileStart`. The file is kept open during the run and a set of values is read from the file each tide. When the end of file is reached, reading starts again at the beginning of the file.

Contents (each line in the file): tidal amplitude value (m), wave period value (s), then a wave height (m) and wave angle value (degrees) for each wavepoint (1 : nWavepoints).

SCAPE works on *origin* angles, i.e. angle that the waves are coming from.

SEALEVELFILE

Provides values for the sea level, indexed by year.

Contents (each line in the file): A Gregorian year value and a sea level value at the start of the year (m from OD).

The file must include a sea level value for every year in the run plus one more year.

BERMWIDTHSFILE

Provides values for the initial width of the berm.

Contents: A value (m) for each shore section (1 : nSections).

BEACHVOLUMESFILE

Provides values for initial beach volumes.

Contents: A value (m^3) for each shore section (1 : nSections).

TALUSVOLUMEFILE

Provides values for the initial talus volume.

Contents: A value (m^3) for each shore section (1 : nSections).

OFFSHOREBARVOLFILE

Provides values for the initial offshore bar volume.

Contents: A value (m^3) for each shore section (1 : nSections).

ROCKSTRENGTHFILE

Provides values for the strength of the platform/cliff material.

Contents: For each cliff element (1 : nce), a value for each shore section (1 : nSections).

BEACHCONTENTFILE

Provides values for initial sand fraction (sediment content) ratio.

Contents: A value (proportion in the range 0.0 to 1.0) for each shore section (1 : nSections).

STARTPROFILESFILE

Provides values for the initial cliff profile.

Contents: For each cliff element (1 : nce), a value (m) for each active shore section (1 : lastActiveSection).

PROFOFFSETSFILE

Provides values for initial section offsets.

Contents: A value (m) for each shore section (1 : nSections).

BASEGEOLOGYFILE

Only required if the horizontal grid is being used (i.e. if `useVAgrid` is .true.) in which case it provides values for the base geology profiles.

Contents: strength value and sediment content value (proportion in the range 0.0 to 1.0), then a count of the number of profile points followed by pairs of Y (m) and Z (m) values for the base geology profile.

This data repeated for each shore section in the model (1 : nSections).

LAYERDATAFILE

Only required if the horizontal grid is being used (i.e. if `useVAgrid` is .true.) in which case it provides values for the layer profiles.

Contents: A count of the number of layers initially present, then for each layer, a strength value and sediment content value (proportion in the range 0.0 to 1.0), followed by a count of the number of profile points followed by pairs of Y (m) and depth (m) values.

This data repeated for each shore section in the model (1 : nSections).

CLIFFDATAFILE

Provides values for each cliff data section.

Contents (each line in the file): cliff height (m), initial mean angle, initial angle standard deviation, mean falling angle, falling angle standard deviation, mean stable angle, stable angle standard deviation and Y section number for each cliff data section.

Note that if cliffs do not extend along the entirety of the model then the number of lines in this file (the number of cliff data sections) will be less than the number of model Y sections.

SEAWALLCONSTRUCTIONYEARSFILE

Provides dates at which the seawalls are installed and optionally provides seawall positions.

Contents: A seawall construction year value or zero, followed by a seawall position value (m) if `seawallMode` is `defineSeawallsByYearAndPosition`, for each shore section (`1 : nSections`).

SEAWALLREMOVALYEARSFILE

Provides dates at which the seawalls are removed.

Contents: A seawall removal year value or zero for each shore section (1 : `nSections`).

GROYNECONSTRUCTIONYEARSFILE

Provides dates at which the groynes are installed.

Contents: A groyne construction year value or zero for each Q point (1 : `nSections` + 1).

GROYNEREMOVALYEARSFILE

Provides dates at which the groynes are removed.

Contents: A groyne removal year value or zero for each Q point (1 : `nSections` + 1).

WAVEEXCLUSIONANGLESFILE

Provides angles from shore normal, beyond which waves are excluded (degrees).

Contents: An `exclusionAngleL` and an `exclusionAngleR` value for each Q point (`1 : nSections + 1`).

Where `exclusionAngleL` is an angle anti-clockwise from shore normal, beyond which waves are excluded (degrees) and `exclusionAngleR` is an angle clockwise from shore normal, beyond which waves are excluded (degrees).

CELERITY TABLES

By default SCAPE+ assumes celerity tables have 5000 depth values and 221 period values and reads the tables from the binary files `clut.bin` and `cglut.bin`.

If `celerityTablesAsTxtFiles` parameter is `.true`. SCAPE+ reads the table dimensions `depthValues` and `periodValues` and reads the celerity tables from the text files `clut.txt` and `cglut.txt`.

CLUT.BIN AND CGLUT.BIN

These files contain look-up tables for wave celerity and wave group celerity in a binary format.

CLUT.TXT AND CGLUT.TXT

These files contain look-up tables for wave celerity and wave group celerity.

For each of `depthValues` water depth values starting at 0.01m and incrementing by 0.01m, SCAPE+ reads `periodValues` real numbers from the files, being the celerity values for `periodValues` wave period values starting at 1.0 secs and incrementing by 0.1 secs. The group of values for each depth should start on a new line in the file.

SHAPEFUNCTIONDRIFT AND SHAPEFUNCTIONEROSION

These files contain shape functions describing cross-shore distributions of alongshore sediment transport and erosion. The erosion shape function was defined by Walkden and Hall (2005).

VELLINGASECTIONSFILE

Identifies the beach sections at which Vellinga effects can apply.

Contains a list of shore section numbers.

4 MODEL OUTPUTS

Some outputs are produced in the same form each time SCAPE+ runs.

Other outputs are influenced by model parameters, as shown in Table 4:

Table 4. Model outputs

Model parameters	Related outputs

`firstYearProfileOutput`
`lastYearProfileOutput`
`firstSectionProfileOutput`
`lastSectionProfileOutput`
`profileOutputTimestep`
	Profile Outputs

`firstYearTransportOutput`	Groyne Factors Sediment Transport Annual Totals Volume Annual Totals

`lowResOutTimestep`
`highResOutStartYear`
`highResOutTimestep`
	Groyne Factors
	Sediment Transport Annual Totals
	Volume Annual Totals
	Fines Volume
	Contour Outputs
	Sea Level Change
	Seawall Active

`velingaActive`	Vellinga Levels

Output files with similar content are grouped in the following sections.

Use the variable name shown in the File unit variable columns in the tables below to search for relevant code and documentation.

As noted above, looking offshore, Q point and shore section indices run right to left (i.e. index 1 is at the right-hand end of the model).

4.1 LOG OUTPUT

The file `log.txt` provides information about the run, the model parameters and metadata related to the output files.

At the head of `log.txt` is the current SCAPE+ version number followed by timeframe information for the run and information on the model grid.

Next, all the SCAPE+ model parameters together with their values (whether defaulted or read from `setup.txt`) are explicitly listed. These parameter settings can be copied into the body of another `setup.txt` file if required.

The output metadata lists the year number for which values will be output. The output files are named and for each the number of values output per year and the units are given where applicable.

If profile outputs are to be produced, the years and section number selection are described.

4.2 SEDIMENT FLUX OUTPUTS

Table 5. Sediment flux outputs

Description	File unit variable File name
Sediment flux at left end of model	`saveSedFluxLeftUnit` `SED_FLUX_L_BOUND.TXT`
Potential sediment flux at left end of model	`savePotSedFluxLeftUnit` `SED_FLUX_POT_L_BOUND.TXT`
Sediment flux at right end of model	`saveSedFluxRightUnit` `SED_FLUX_R_BOUND.TXT`
Potential sediment flux at right end of model	`savePotSedFluxRightUnit` `SED_FLUX_POT_R_BOUND.TXT`

Contents: Count of the number of values per year (generally 706). Flux value (m^3) for each tide, repeated for each year in the run.

4.3 CONTOUR OUTPUTS

Table 6. Contour outputs

Description	File unit variable File name
Rock contour	`saveRockContourUnit` `CONTOUR_ROCK.TXT`
Shore contour	`saveShoreContourUnit` `CONTOUR_SHORE.TXT`

For the rock contour, at the end of every year the value of the `cliffy` element at the outputContourLevel is saved for all beach sections.

For the shore contour, at the end of every year the value of the maximum of the `cliffy` element at the outputContourLevel and the beachy element at the outputContourLevel is saved for all beach sections.

Contents: Count of the number of values per year (`nSections`). Value (m) for all beach sections (1 : `nSections`) repeated for each year in the run.

4.4 SEDIMENT TRANSPORT ANNUAL TOTALS OUTPUTS

Table 7. Sediment transport annual totals outputs

Description	File unit variable File name
Annual sediment transport	`saveSedTransAnnUnit` `SED_TRANS_ANN.TXT`
Potential annual sediment transport	`savePotTransAnnUnit` `SED_TRANS_POT_ANN.TXT`

Contents: Count of the number of values per year (`nSections` + 1). Value (m^3) for all Q points (1 : `nSections` + 1) repeated for each year from model input parameter `firstYearTransportOutput` inclusive.

4.5 VOLUME ANNUAL TOTALS OUTPUTS

Table 8. Volume annual totals outputs

Description	File unit variable File name
Annual beach addition	`saveBeachAddAnnUnit` `LIBERATED_SED_BEACH.TXT`
Annual beach volume	`saveAnnBvolUnit` `VOLUME_B_ANN_AVE.TXT`

Contents: Count of the number of values per year (`nSections`). Value (m^3) for all beach sections (1 : `nSections`) repeated for each year from model input parameter `firstYearTransportOutput` inclusive.

4.6 PROFILE OUTPUTS

Table 9. Profile outputs

Description	File unit variable File name
Rock profiles	`saveRockProfilesUnit` `ROCK_PROFILES.TXT`
Beach profiles	`saveBeachProfilesUnit` `BEACH_PROFILES.TXT`

Profile outputs (see Table 9) are produced only if the model input parameter `firstYearProfileOutput` is set to a value in the range `firstYear` to `lastYear`. Profile outputs are further controlled by these model input parameters:

`lastYearProfileOutput`

`firstSectionProfileOutput`

`lastSectionProfileOutput`

`profileOutputTimestep`

Rock profiles contain `cliffy` values for all vertical cliff elements.

Beach profiles contain beachy values only for active beach cliff elements.

Contents: A number of values taken from the model input parameters:

Number of vertical cliff elements

Number of beach sections for which profile is output

Number of years for which profile is output

Vertical element height (m)

Beach section width (m)

Number of years between profile results outputs

First section for which profile results are output

Last section for which profile results are output

First year for which profile results are output

Last year for which profile results are output

Then, for each year for which profile output is required, for each shore section for which profile output is required, the value of the elevation (to national datum) of the lowest element in the output, followed by the vertical cliff element values (m) from lowest to highest.

4.7 GROYNE FACTORS OUTPUT

Table 10. Groyne factors output

Description	File unit variable File name
Groyne Factors	`saveGroyneEffectUnit` `GROYNE_FACTORS.TXT`

Contents: Count of the number of values per year (`nSections` + 1). Value for all Q points (1 : `nSections` + 1) repeated for each year from model input parameter `firstYearTransportOutput` inclusive. Values show the effect of groyne at a Q point where zero means "no effect" and 1 means "completely blocks flow", averaged over the year.

4.8 FINES VOLUME OUTPUT

Table 11. Fines volume output

Description	File unit variable File name
Fines volume	`saveFinesVolUnit` `LIBERATED_SED_FINES.TXT`

Total fines addition to the beach over a year.

Contents: Count of the number of values per year, followed by the value (m^3) repeated for each year for which output is required.

4.9 SEAWALL ACTIVE OUTPUT

Table 12. Seawall active output

Description	File unit variable File name
Seawall active	`saveSeawallActiveUnit` `SEAWALL_PRESENCE.TXT`

Contents: Count of the number of values per year (`nSections`). Value (0 meaning seawall inactive or 1 meaning seawall active) for all beach sections (1 : `nSections`) repeated for each year for which output is required.

4.10 VELLINGA LEVELS OUTPUT

Table 13. Vellinga levels output

Description	File unit variable File name
Vellinga levels	`saveVellLevelsUnit` `VELL_LEVELS.TXT`

Contents: Count of the number of Vellinga sections. Vellinga level value for each Vellinga section repeated for each year for which output is required.

BIBLIOGRAPHY

- Walkden, M., Johnson, A., Cooper, N. 2014. Cliff and Shore Erosion under Accelerating Sea Level Rise: Results of Case Study Analysis Joint Defra/EA Flood and Coastal Erosion Risk Management R&D Programme, Report – SC120017/03.
- Ashton, A.D., Walkden, M.J. and Dickson, M.E., 2011. Equilibrium responses of cliffed coasts to changes in the rate of sea level rise. *Marine Geology*, *284*(1), pp.217-229.
- Carpenter, N., Dickson, N., Nicholls, R., Powrie, W., and Walkden, M. (2014) Investigating the Effects of Varied Lithology on Soft Cliff Geomorphology and Recession Rates. Marine Geology
- Chini, N., Stansby, P., Walkden, M., Hall, J., Wolf, J., Lowe, J., Nicholls, R. (2010). Modelling Long Term Implication of Climate Change Projection on Shore Morphology of North Norfolk, UK. International Conference on Coastal Engineering. Beijing.
- Cooper, N., Brew, D., Williams, P., Balson, P., Pearson, S., Moses, C., Williams, R. and Charman, R., 2007. Erosion of Cohesive Shore Platforms. Coastal management. January 2007, 69-78, DOI: 10.1680/cm.35683.0007
- Dawson, R.J., Dickson, M.E., Nicholls, R.J., Hall, J.W., Walkden, M.J., Stansby, P.K., Mokrech, M., Richards, J., Zhou, J., Milligan, J. and Jordan, A., 2009. Integrated analysis of risks of coastal flooding and cliff erosion under scenarios of long term change. *Climatic Change*, *95*(1-2), pp.249-288.
- Defra/EA (2007), *Understanding and Predicting Beach Morphological Change Associated with the Erosion of Cohesive Shore Platforms*. Joint Defra/EA Flood and Coastal Erosion Risk Management R&D Programme, R&D Technical Report FD1926/TR
- Dickson, M.E., Walkden, M.J.A., Hall, J.W., 2007. Systemic impacts of climate change on an eroding coastal region over the twenty-first century. Climatic Change 84 (2), 141-166.
- Hanson, H. and Kraus, N.C., 1989.*GENESIS: Generalized Model for Simulating Shoreline Change. Report 1. Technical Reference* (No. CERC-TR-89-19-1). COASTAL ENGINEERING RESEARCH CENTER VICKSBURG MS.
- Nairn, R.B. and Southgate, H.N., 1993. Deterministic profile modelling of nearshore processes. Part 2. Sediment transport and beach profile development. *Coastal Engineering*, *19*(1), pp.57-96.
- van Rijn, L.C., 2014. A simple general expression for longshore transport of sand, gravel and shingle. *Coastal Engineering*,*90*, pp.23-39.
- Vellinga, P., 1986.*Beach and dune erosion during storm surges* (Doctoral dissertation, TU Delft, Delft University of Technology).
- Walkden, M.J.A. and Hall, J.W., 2005. A predictive mesoscale model of the erosion and profile development of soft rock shores. *Coastal Engineering*,*52*(6), pp.535-563.
- Walkden, M.J. and Hall, J.W., 2011. A mesoscale predictive model of the evolution and management of a soft-rock coast. *Journal of Coastal Research*, *27*(3), pp.529-543.
- Walkden, M.J and Rossington, K., (2009). Characterisation and prediction of large scale, long-term change of coastal geomorphological behaviours: Proof of concept modelling. Joint Defra/EA Flood and Coastal Erosion Risk Management R&D Programme, Science Report: SC060074/PR1.
- Walkden, M.J. and Dickson, M.E., Equilibrium erosion of soft rock shores with a shallow or absent beach under increased sea level rise. *Marine Geology* 251.1 2008: 75-84.
- Walkden, M., Dickson, M., Thomas, J., and Hall, J. (2015). *Simulating the Shore and Cliffs of North Norfolk*. Chapter 7 of *Broad scale coastal simulation: new techniques to understand*

and manage shorelines in the third millennium, Springer (editors: Nicholls, R., Dawson, R., and Day, S).

- Walkden, M., Watson, G, Johnson, A., Heron, E., and Tarrant, O. (2015b). Coastal Catch-up Following Defence Removal at Happisburgh. Paper presented at the ICE Coastal Management conference, 7-9th September, Amsterdam.
- Walkden, M., Payo, A., Barnes, J., and Burningham, H., (2015). *Modelling the response of coupled barrier and cliff systems to sea level rise*. Proceedings of Coastal Sediments, San Diego.

Appendix A

LICENSE

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to “keep intact all notices”.
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's “contributor version”.

A contributor's “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this

License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

Appendix B

RUNNING SCAPE+ WITHIN A FLUIDEARTH COMPOSITION

APPENDIX B-1

RUNNING SCAPE+ WITH FLUIDEARTH

Running SCAPE with OpenMI

SCAPE Folder Structure

The top level folder “SCAPE” has all the source files, a Silverfrost FORTRAN project file and a Makefile (for use with other compilers) to build and run the freestanding SCAPE application “RUN_SCAPE.EXE”. The folder also includes RUN_SCAPE.EXE built using gFORTRAN for Windows.

Files	Description
*.f90, *.f95	FORTRAN source files
SCAPE.ftn95p	Silverfrost FORTRAN project file
Makefile	gFORTRAN make file for RUN_SCAPE.EXE (can be edited to suit other FORTRAN compilers)
RUN_SCAPE.EXE	gFORTRAN make of RUN_SCAPE.EXE.
setup.txt and other *.txt, *.bin files	Setup file and data files for a 100 year example run

ENGINE sub-folder

This folder has all the additional or changed files required to build the ENGINE.dll object that provides SCAPE functionality in the OpenMI context. This build also uses all the FORTRAN source files in the parent folder, except the file exceptions.f95. The folder also includes ENGINE.dll built using gFORTRAN for Windows.

Files	Description
fluidearth_support.f95 FluidEarth2_DllExports.f90 FluidEarth2_Exceptions.f90	Additional FORTRAN source files
exceptions.f95	FORTRAN source file used instead of the file of the same name in the parent folder
ENGINE.ftn95p	Silverfrost FORTRAN project file
Makefile script	gFORTRAN make file and script file for ENGINE.dll (can be edited to suit other FORTRAN compilers)
ENGINE.dll	gFORTRAN make of ENGINE.dll

Component sub-folder

This folder has all the additional files required to build the CREATE_FILES.EXE application that creates the .omi and .fcs files that describe a SCAPE component to OpenMI. This build also uses some of the FORTRAN source files in the parent folder. The folder also includes CREATE_FILES.EXE built using gFORTRAN for Windows.

Files	Description
component.f95 main.f95 support.f95	Additional FORTRAN source files
Component.ftn95p	Silverfrost FORTRAN project file
Makefile	gFORTRAN make file for CREATE_FILES.EXE (can be edited to suit other FORTRAN compilers)
CREATE_FILES.EXE	gFORTRAN make of CREATE_FILES.EXE

template.omi template.fcs	Templates used to create the SCAPE.omi and SCAPE.fcs xml files written by CREATE_FILES.EXE.
component.txt	Parameters needed by CREATE_FILES.EXE.

Engine_Wrapper sub-folder

A C# .net project that builds the wrapper dll for FORTRAN 90 builds of the SCAPE ENGINE.dll dll together with a pre-built copy of the dll.

Engine_Wrapper2 sub-folder

A C# .net project that builds the wrapper dll for FORTRAN 2003 builds of the SCAPE ENGINE.dll dll together with a pre-built copy of the dll.

Other Software Required

A build or installation of the FluidEarth composition builder and runner Pipistrelle.

A build or installation of the FluidEarth SDK.

Building the SCAPE Engine

Using Silverfrost FORTRAN

Open the project ENGINE.ftn95p and build either Debug Win32 or Release Win32.

Copy Engine_Wrapper.dll from the Engine_Wrapper folder to the folder where ENGINE.dll has been built.

Using gFORTRAN

Using the Makefile, type (for example) make in a command window open at the ENGINE folder. The source files required from the parent folder will be copied to the ENGINE folder before the rest of the make runs to build ENGINE.dll.

Copy Engine_Wrapper.dll from the Engine_Wrapper2 folder to the folder where ENGINE.dll has been built.

Building the SCAPE Component Creator

Using Silverfrost FORTRAN

Open the project Component.ftn95p and build either Debug Win32 or Release Win32.

Using gFORTRAN

Using the Makefile, type (for example) make in a command window open at the Component folder. The source files required from the parent folder will be copied to the Component folder before the rest of the make runs to build CREATE_FILES.EXE.

Creating a SCAPE Model/Component

Create a folder to contain the model/component and ensure that all these files are present in that folder:

Files	Description
setup.txt and all the model *.txt and *.bin files	Model setup and data files

component.txt	File containing parameters required by CREATE_FILES.EXE

The parameters in component.txt are:

Parameter	Description
TEMPLATESPATH	The next line gives the path to the location of the files template.omi and template.fcs
ENGINEWRAPPERPATH	The next line gives the path to the file Engine_Wrapper.dll
FLUIDEARTHPATH	The next line gives the path to the file FluidEarth2_Sdk.exe
SEAWARDBOUNDARY	The next line gives the Seaward Boundary offset (Y-value) (m) – default is 1000.0
WORLDCOORDSID	The next line gives an identifier string for the world coordinates – default is an empty string, meaning there is to be no conversion to world coordinates
LOCALCOORDSX	The next line gives the X-value of the local coords origin in world cords (m) – only required if WORLDCOORDSID is not empty
LOCALCOORDSY	The next line gives the Y-value of the local coords origin in world cords (m) – only required if WORLDCOORDSID is not empty

An example of the contents of a component.txt file might be:

```
TEMPLATESPATH
C:\FortranProjects\SCAPE\Component\
ENGINEWRAPPERPATH
C:\FortranProjects\SCAPE\Engine_Wrapper\Engine_Wrapper.dll
FLUIDEARTHPATH
C:\Source\FluidEarth2_Sdk\bin\x86\Debug\FluidEarth2_Sdk.exe
```

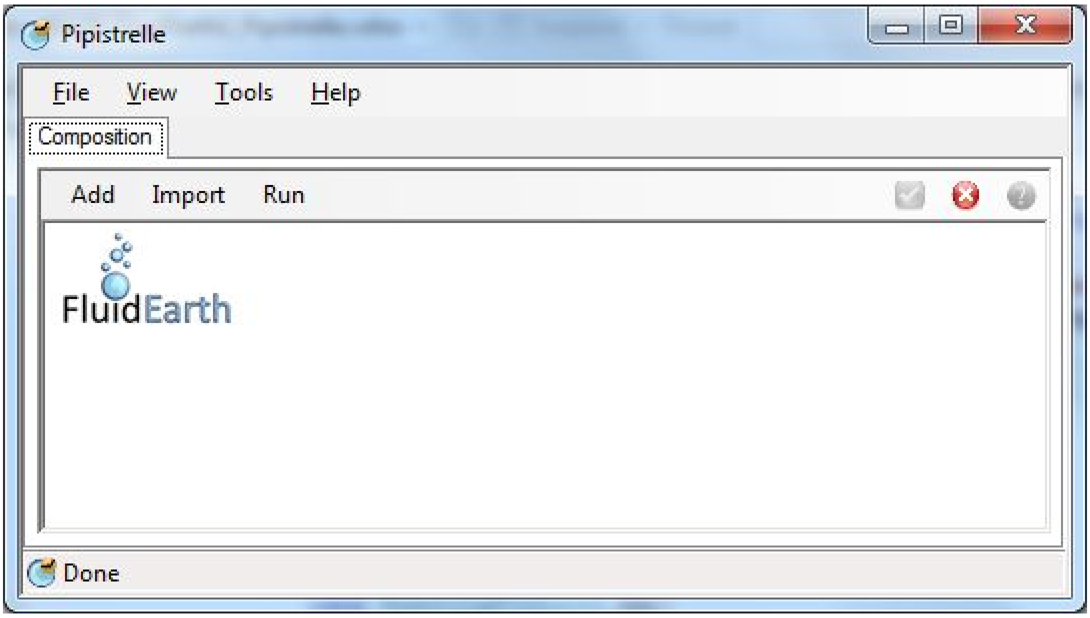
Open a command window and change the current directory to the model/component folder.

If you want to give the component the default caption of "SCAPE", just run CREATE_FILES.EXE with no arguments.

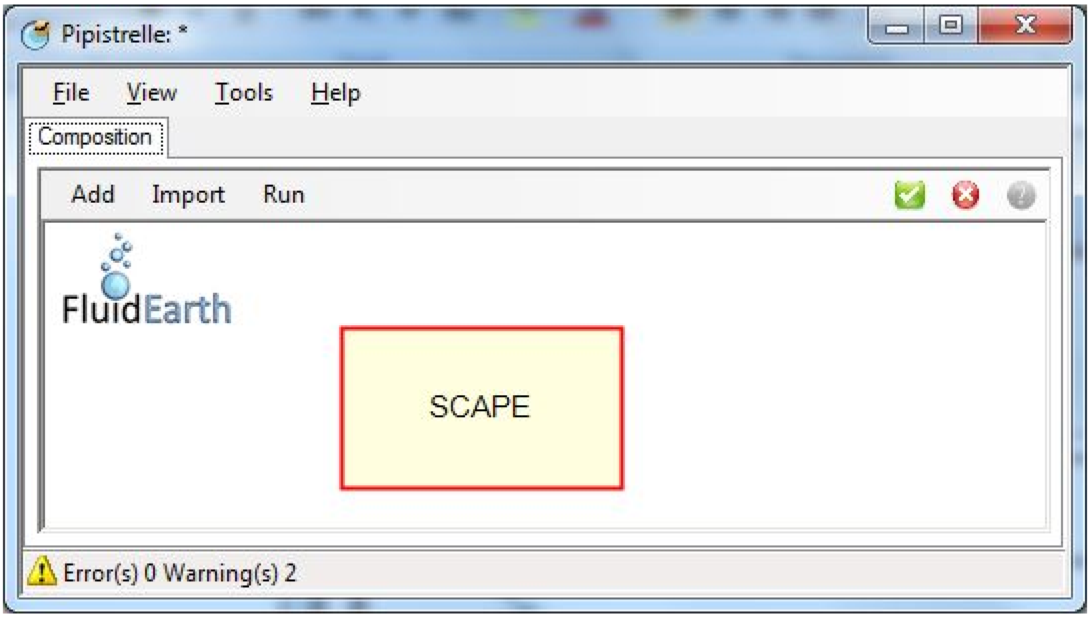
If you want to give the component another caption, e.g. because you will be running more than one instance of the SCAPE component in the same OpenMI composition, run CREATE_FILES.EXE with the required caption as a command-line argument.

Running SCAPE in OpenMI

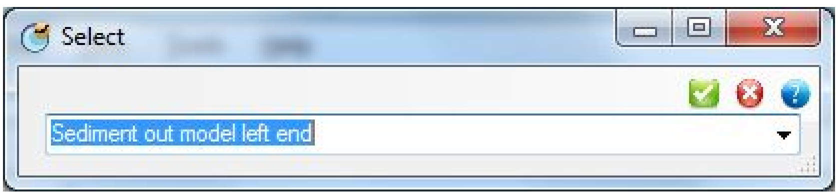
Start the FluidEarth OpenMI composition runner, Pipistrelle:



Click on “Add”, then click on “Component ...” and open the .omi file for the SCAPE component:

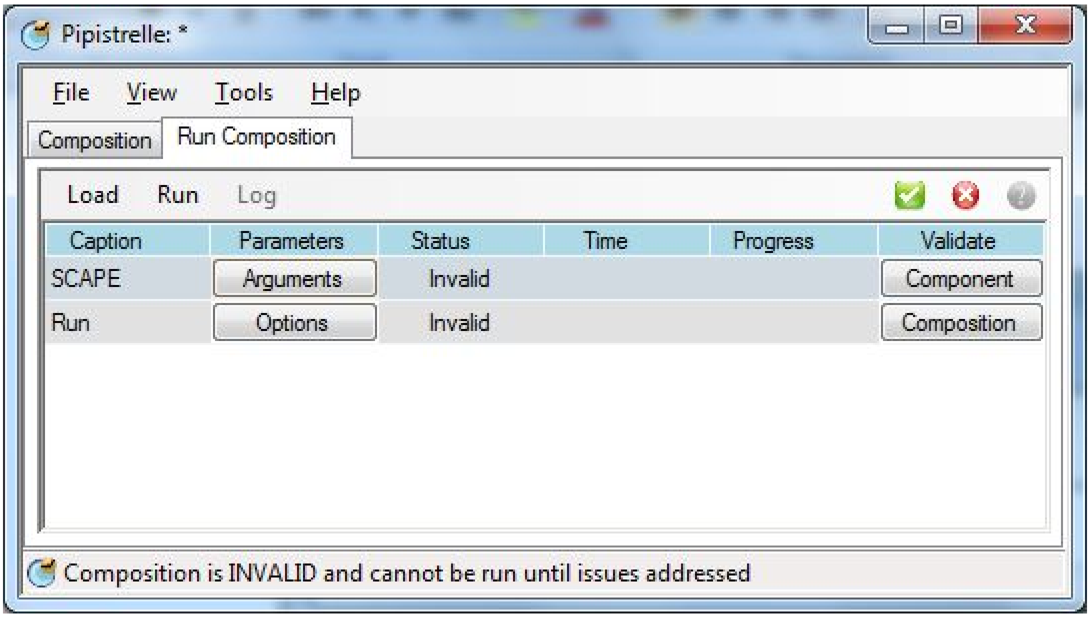


Right-click on the SCAPE component then click on “Set as “Get Values Call” ...” and select an output for the run to pull on:

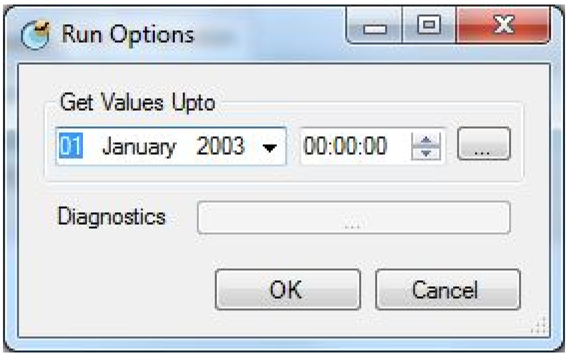


Click on the green tick, to close the Selection dialog.

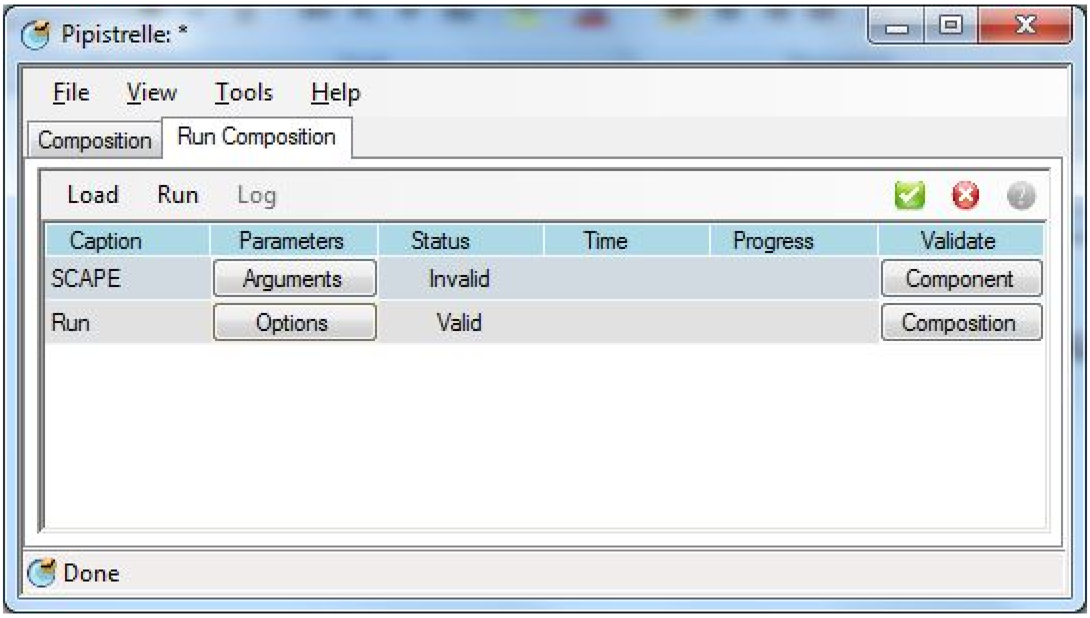
Click on “Run” then click on “Pull Run”:



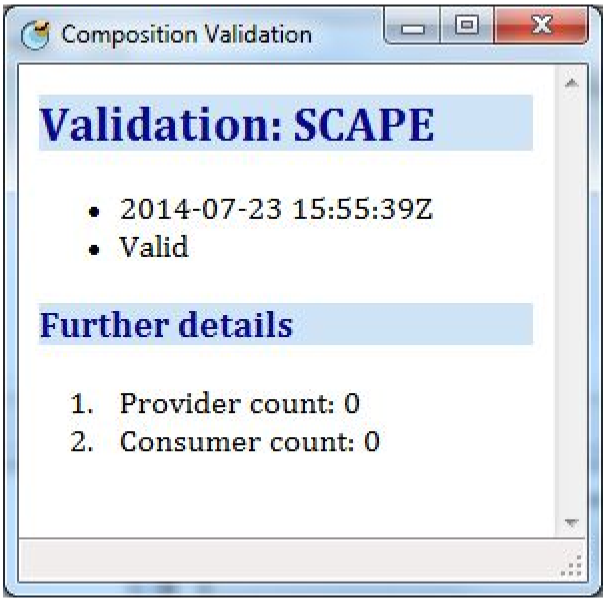
Click on the “Options” button and set a suitable value for the “Get Values Upto” fields. The choice of a suitable year value will depend on the SCAPE model embedded in the component and what you want to do with it. Setting the date to 1st January and the times to 00:00:00 is recommended for SCAPE:



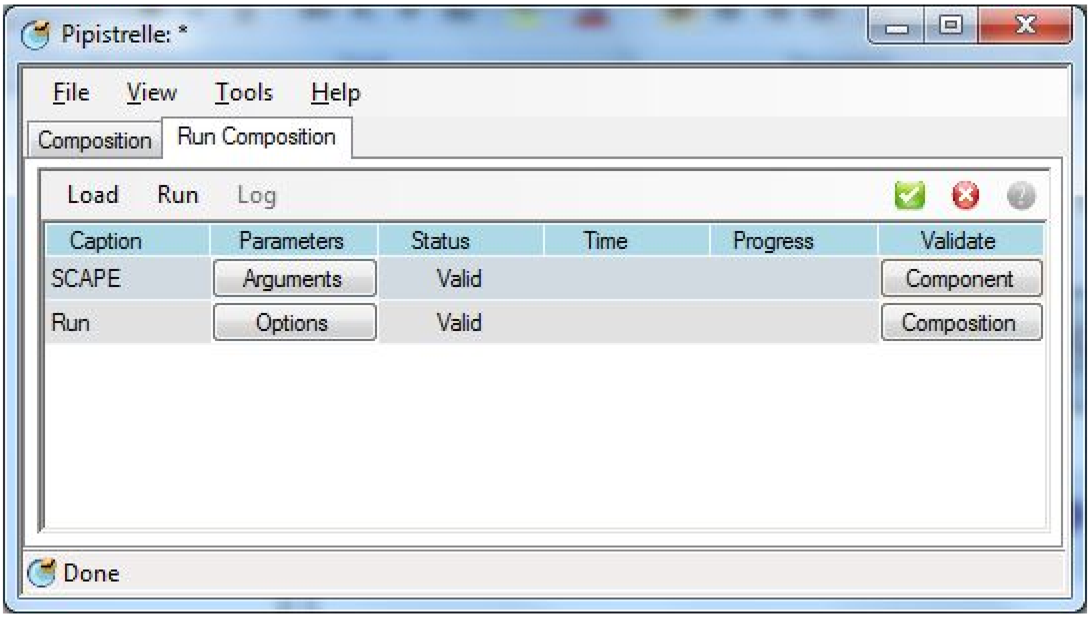
OK the Run Options dialog and you should see:



Click on the “Component” button in the SCAPE row and check the validation:



Close the Composition Validation window and the composition should be ready to run:



Click on “Run” then click on “Start” and you will be prompted to save the .CHI file where Pipistrelle stores the composition. Complete the Save and the run should begin.

Note that when Pipistrelle runs the SCAPE Component it will invoke Engine_wrapper.dll. This in turn will invoke ENGINE.dll and any code upon which it depends. To ensure that ENGINE.dll and any code upon which it depends can be found at run-time, such code should be co-located with Engine_wrapper.dll.

APPENDIX B-2

CREATING A FLUIDEARTH COMPOSITION

Creating a SCAPE+ - MESO_i – SCAPE+ Composition

This document goes through all the steps needed to create a typical FluidEarth composition involving two SCAPE+ models connected to a MESO_i model.

Assumed Setup

The three components to be linked have already been set up in three folders named Bawdsey, MESO_i and Felixstowe. Each of these folders contains a .omi file and a .fcs file defining the component and all the necessary model files.

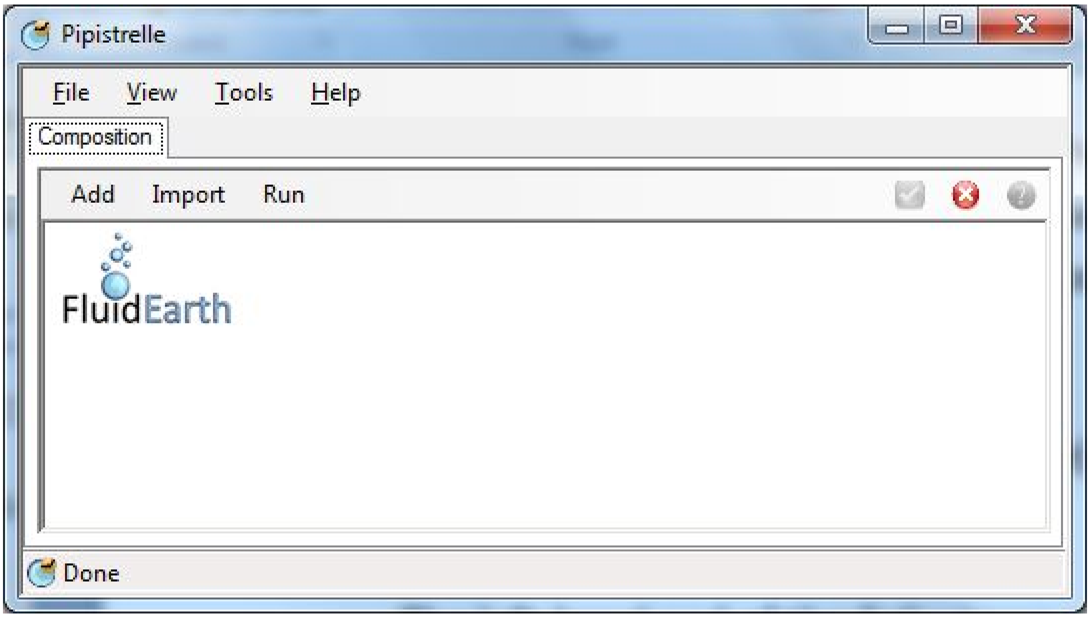
Required Composition

The right-hand end (looking offshore) of the Bawdsey model is to exchange data with the Updrift end of the MESO_i model.

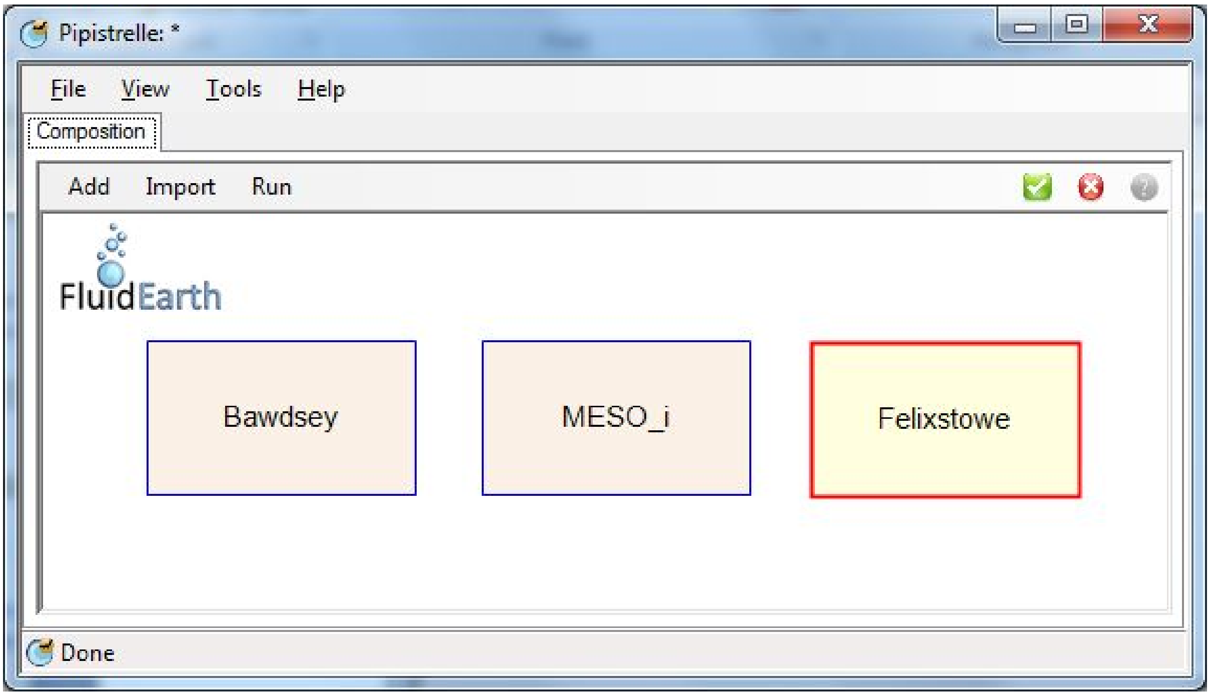
The left-hand end of the Felixstowe model is to exchange data with the Downdrift end of the MESO_i model.

Steps to Create the Composition

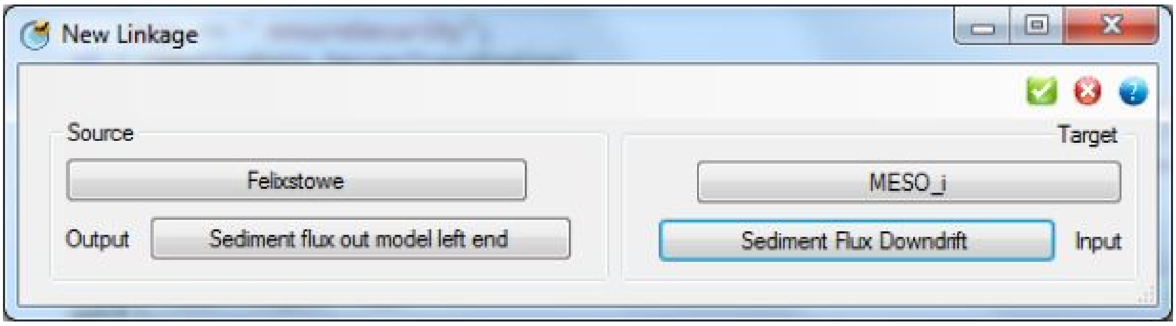
Run Pipistrelle, the FluidEarth composition creation and editing tool.



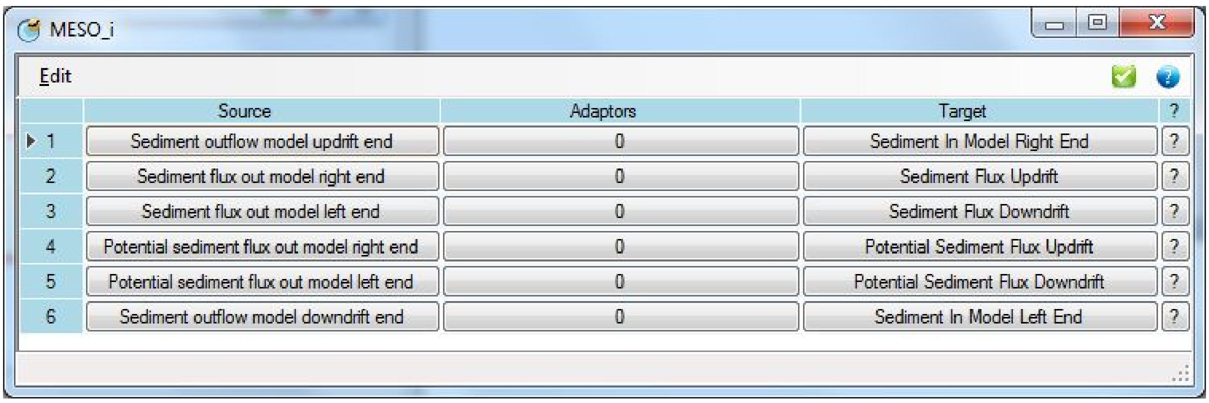
Add each component in turn by clicking the Component... command on the Add menu of the Composition tab and navigating to the .omi file. Arrange the components artistically in the Composition view.



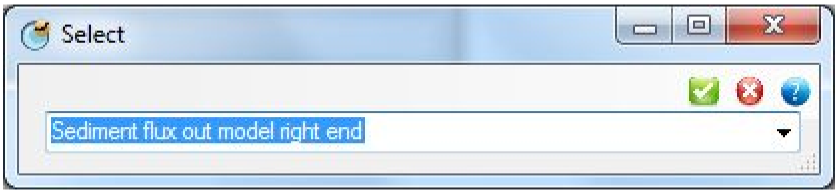
Add each required linkage in turn by clicking the Connection... command on the Add menu of the Composition tab and selecting the Source and Target components and input and output for the linkage, for example:



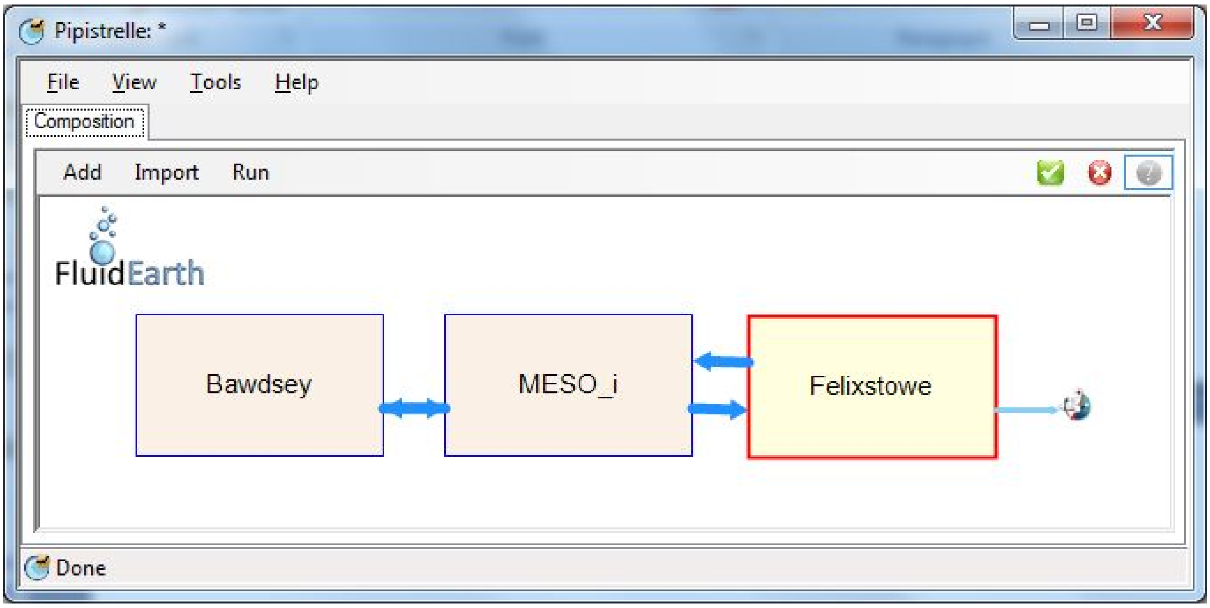
When all linkages have been made, select the MESO_i component and click on the Connectivity command in the popup menu to see the linkages:



Select the Bawdsey or Felixstowe component and click on the Set as “Get Values Call”... in the popup menu to select an unused output for the run to pull on, for example:



The composition should now look something like this:

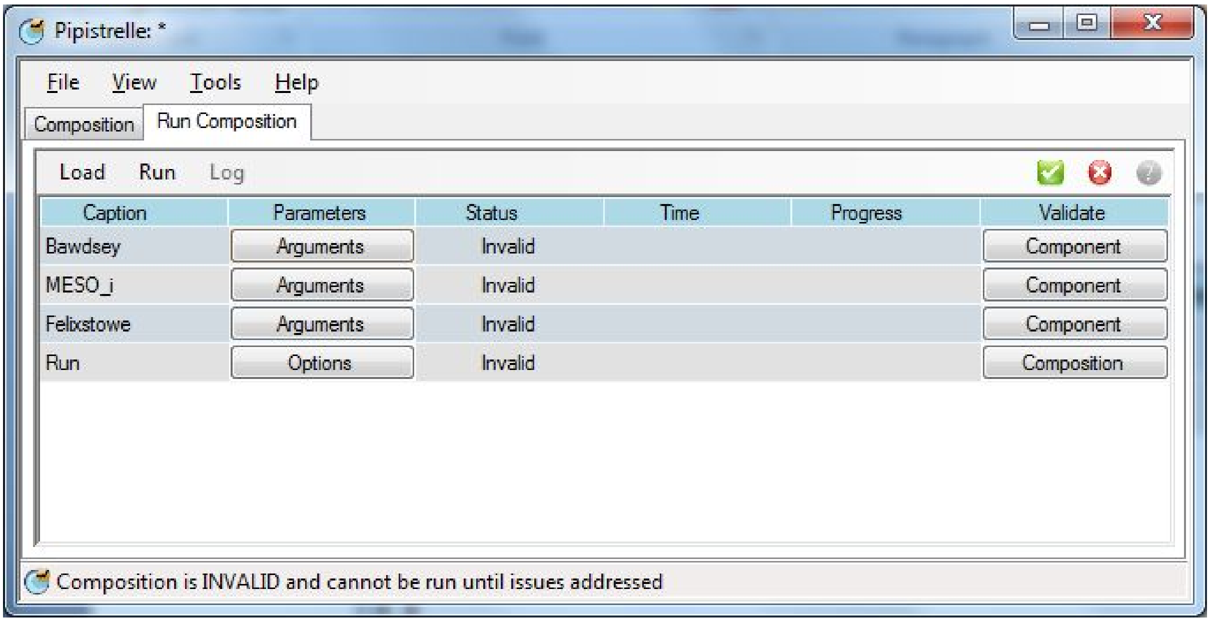


Before going further it may be wise to look at the Time Horizon properties of each of the three components and check them for consistency with what you have in mind for the run. Bear in mind that data exchange between two components will only start when both components are at or after their Time Horizon start time. Output of results values to files will start at component Time Horizon start time.

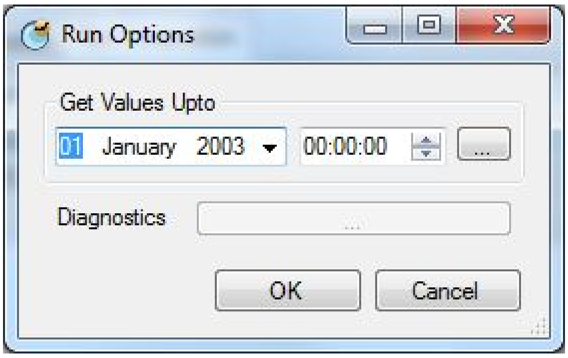
For SCAPE+ components, the Time Horizon start time can be later than the start year built into the model, in which case the model will have an initial “run up” period during which no results are output.

For MESO_i components, the model simply starts running and writing out results at the Time Horizon start time.

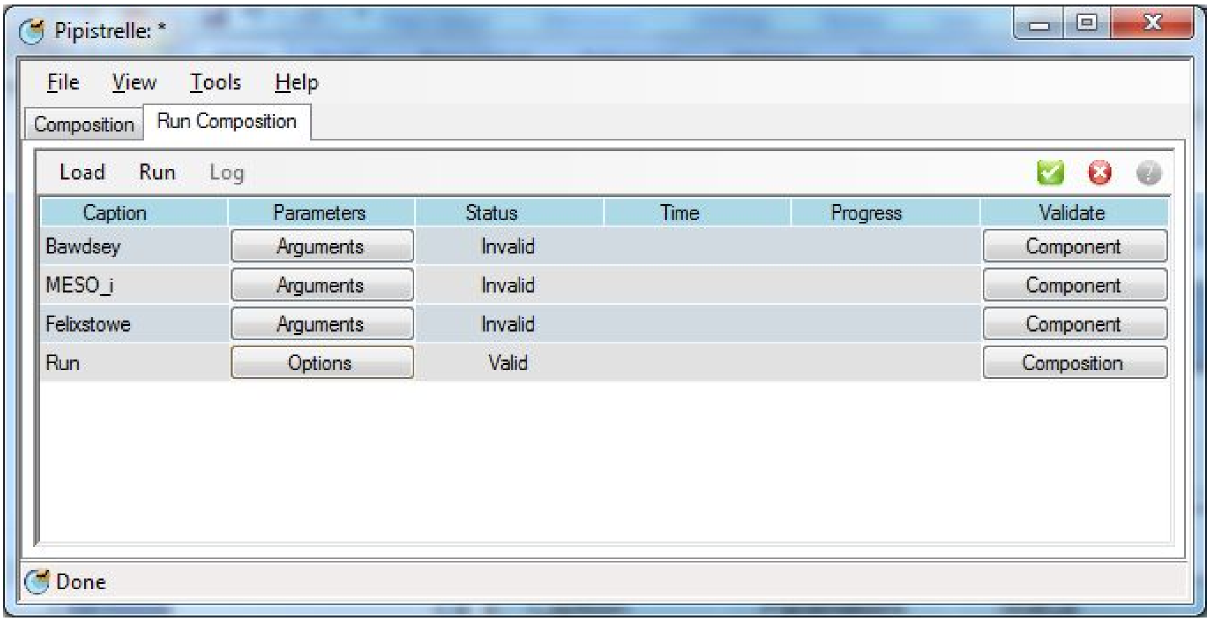
Click on “Run” then click on “Pull Run”:



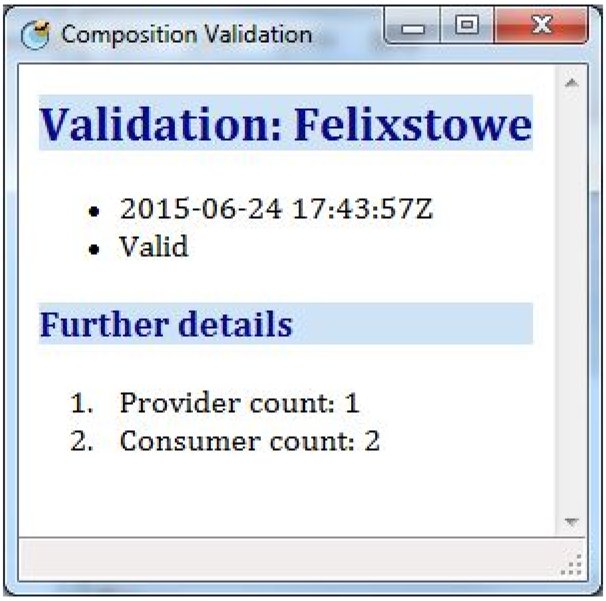
Click on the “Options” button and set a suitable value for the “Get Values Upto” fields. The choice of a suitable year value will depend on the run you want to do. Setting the date to 1st January and the times to 00:00:00 is recommended for SCAPE runs:



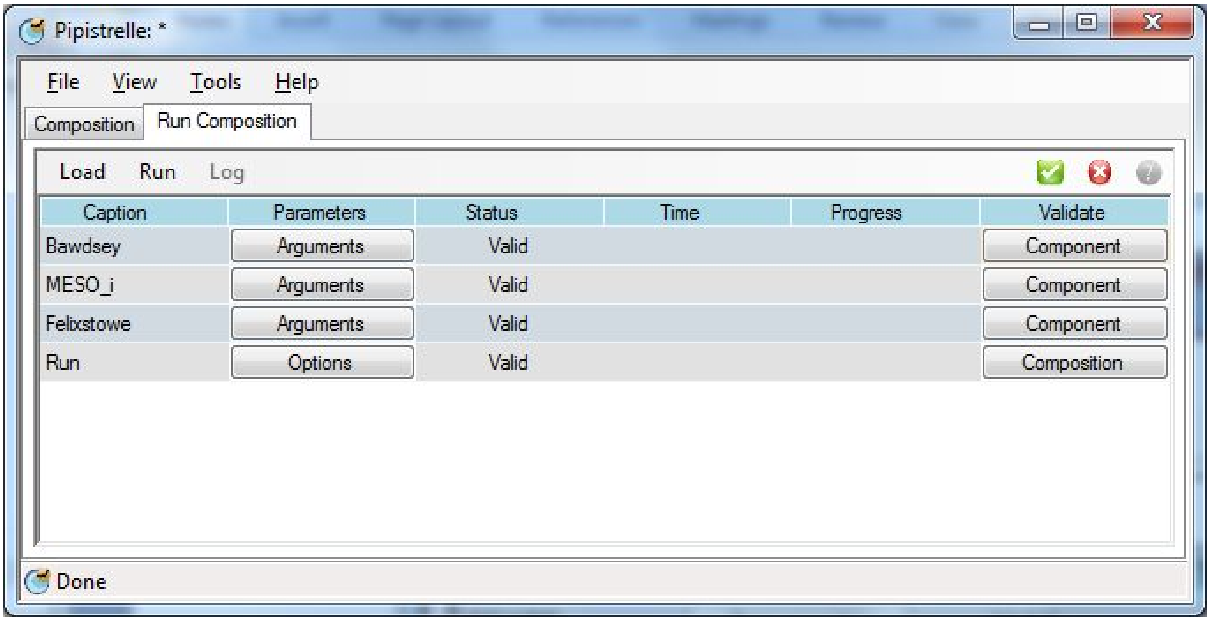
OK the Run Options dialog and you should see:



Click on the “Component” buttons in the Baswdsey, MESO_i and Felixstowe rows and check the validations, for example:



The composition should now be ready to run:



Click on “Run” then click on “Start” and you will be prompted to save the .CHI file where Pipistrelle stores the composition. Complete the Save and the run should begin.