Community Land Model

Release 1.0.0

Grant Johnson, Kevin Yi, Henry Stone

Dec 16, 2022

CONTENTS:

CHAPTER

ONE

COMMUNITY_LAND_MODEL

1.1 DIAGNOSTICS package

1.1.1 Submodules

1.1.2 DIAGNOSTICS.diagnostics module

Routines for analysing output data

#Requires linking to the defintions:

DIAGNOSTICS.diagnostics.**diagnostics**()

DIAGNOSTICS.diagnostics.**plot_general**(*X*, *Y*, *str_val_x*, *str_val_y*)

1.1.3 Module contents

1.2 EXAMPLES package

1.2.1 Submodules

1.2.2 EXAMPLES.input_script module

Main Project loop: Inputs: None Outputs: None

#Requires linking to the definitions: initialize_data, run_clm

class EXAMPLES.input_script.**clm_parameters**

Bases: object

EXAMPLES.input_script.**main**()

1.2.3 Module contents

1.3 SRC package

1.3.1 Submodules

1.3.2 SRC.initialization module

Initialization routines: Inputs: structures Parameters Outputs: structures App, State, Grid

#Requires linking to the defintions:

class SRC.initialization.**clm_app**(*parameters*)

Bases: object

load()

try load self.name.txt

save()

save class as self.name.txt

class SRC.initialization.**clm_grid**(*parameters*)

Bases: object

load()

try load self.name.txt

save()

save class as self.name.txt

time(*i*)

class SRC.initialization.**clm_state**(*parameters*, *Grid*)

Bases: object

load()

try load self.name.txt

save()

save class as self.name.txt

class SRC.initialization.**evaporation**(*parameters*, *NT*, *Grid*)

Bases: object

SRC.initialization.**initialize_data**(*parameters*)

class SRC.initialization.**input_data**(*parameters*, *NT*)

Bases: object

class SRC.initialization.**radiation**(*parameters*, *NT*)

Bases: object

class SRC.initialization.**sensible_heat**(*parameters*, *NT*, *Grid*)

Bases: object

1.3.3 SRC.model_absorbed_emitted_radiation module

model_absorbed_emitted_radiation: Inputs: State, Grid, i Outputs: None

#Requires linking to the defintions:

SRC.model_absorbed_emitted_radiation.**run_absorbed_emitted_radiation_model**(*State*, *Grid*, *App*, *i*)

1.3.4 SRC.model_albedo module

Albedo model: Inputs: State, Grid, App, i Outputs: None

#Requires linking to the defintions: TABLES and METHODS subdirectories

SRC.model_albedo.**canopy**(*State*, *Grid*, *App*, *i*)

SRC.model_albedo.**canopy_model**(*State*, *Grid*, *App*, *i*, *j*)

SRC.model_albedo.**lambda_val**(*d*)

SRC.model_albedo.**optical_params**(*State*, *i*, *j*)

SRC.model_albedo.**run_albedo_model**(*State*, *Grid*, *App*, *i*)

SRC.model_albedo.**solar**(*State*, *Grid*, *App*, *i*)

SRC.model_albedo.**solar_zenith_angle**(*State*, *Grid*, *App*, *i*)

1.3.5 SRC.model_evaporation module

Evaporation model: Inputs: State, Grid, i Outputs: None

#Requires linking to the defintions:

SRC.model_evaporation.**MO_length**(*State*, *Grid*, *App*, *i*)

SRC.model_evaporation.**U_av**(*State*, *Grid*, *App*, *i*)

SRC.model_evaporation.**air_res**(*State*, *Grid*, *App*, *i*)

SRC.model_evaporation.**canopy_specific_humidity**(*State*, *Grid*, *App*, *i*)

SRC.model_evaporation.**friction_velocity**(*State*, *Grid*, *App*, *i*, *psi*)

L = MO_length(State,Grid,App,i)[1] Ksi = MO_length(State,Grid,App,i)[2] psi = psi(ksi)

SRC.model_evaporation.**humidity_ratio**(*State*, *Grid*, *App*, *i*, *psi*)

L = MO_length(State,Grid,App,i)[1] Ksi = MO_length(State,Grid,App,i)[2] psi = psi(ksi)

SRC.model_evaporation.**latent_heat_water_vapor_flux**(*State*, *Grid*, *App*, *i*)

SRC.model_evaporation.**psi**(*ksi*)

SRC.model_evaporation.**q_sat**(*State*, *Grid*, *App*, *i*)

SRC.model_evaporation.**r_b**(*State*, *Grid*, *App*, *i*)

SRC.model_evaporation.**ra**(*State*, *Grid*, *App*, *i*)

SRC.model_evaporation.**ra_prime**(*State*, *Grid*, *App*, *i*)

SRC.model_evaporation.**run_evaporation_model**(*State*, *Grid*, *App*, *i*)

SRC.model_evaporation.**temperature_ratio**(*State*, *Grid*, *App*, *i*, *psi*)

L = MO_length(State,Grid,App,i)[1] Ksi = MO_length(State,Grid,App,i)[2] psi = psi(ksi)

SRC.model_evaporation.**vegetation_water_vapor_flux**(*State*, *Grid*, *App*, *i*)

SRC.model_evaporation.**water_vapor_flux**(*State*, *Grid*, *App*, *i*)

1.3.6 SRC.model_sensible_heat_flux module

Sensible heat flux model: Inputs: State, Grid, i Outputs: None

#Requires linking to the defintions:

SRC.model_sensible_heat_flux.**canopy_air_temperature**(*State*, *Grid*, *App*, *i*)

SRC.model_sensible_heat_flux.**run_sensible_heat_flux_model**(*State*, *Grid*, *App*, *i*)

SRC.model_sensible_heat_flux.**sensible_heat_flux_tot**(*State*, *Grid*, *App*, *i*)

SRC.model_sensible_heat_flux.**sensible_heat_flux_v**(*State*, *Grid*, *App*, *i*)

1.3.7 SRC.run_routine module

Run routine, main time loop: Inputs: structures App, State, Grid Outputs: None (Calls: save to file)

#Requires linking to the defintions: file_saves, modules: run_XX

SRC.run_routine.**run_clm**(*App*, *State*, *Grid*)

1.3.8 Module contents

1.4 TESTS package

1.4.1 Submodules

1.4.2 TESTS.test_albedo module

Test for verifying the albedo module

#Requires linking to the defintions: SRC/model_albedo.py

TESTS.test_albedo.**test_albedo**()

1.4.3 TESTS.test_time_integration module

Test for verifying the integration_schemes module

#Requires linking to the defintions: SRC/METHODS/time_integration.py

TESTS.test_time_integration.**test_integration_schemes**()

1.4.4 Module contents

1.5 noxfile module

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

d

e

s

t

INDEX

S

T

U

V

W