

```

In [ ]: from aeolus.model import um
from aeolus.model import um_stash

import warnings
from pathlib import Path

import iris
import iris.cube
import iris.quickplot as qplt
import matplotlib.pyplot as plt
import numpy as np

In [ ]: from aeolus.synthobs import (
    calc_stellar_flux,
    calc_transmission_spectrum,
    calc_transmission_spectrum_day_night_average,
    read_normalized_stellar_flux,
    read_spectral_bands,
)

In [ ]: warnings.filterwarnings("ignore", module="iris")
plt.rcParams["mathtext.default"] = "regular"

In [ ]: iris.FUTURE.datum_support = True

In [ ]: TEST_DIR = Path.cwd().parent / "Outputs_u-cx738"
print(TEST_DIR)
/home/soumik/Documents/Dlibya/Thesis_work_Jayesh/Unified_Model_GCM/Outputs_u-cx738

In [ ]: data_file = TEST_DIR / "transm_files" / "transm_stash_1755"

planet_transm_test = iris.load_cube(data_file)
print(planet_transm_test)

Stash code = 1755 / (unknown) (t: 144; pseudo: 260; latitude: 90; longitude: 144)
Dimension coordinates:
  t                x                -                -                -
  pseudo           -                x                -                -
  latitude         -                -                x                -
  longitude        -                -                -                x
Attributes:
  date             '13/05/04'
  history          'Sun Jul 23 18:39:15 BST 2023 - XCONV V1.93 13-October-2015'
  name             'unspecified'
  source          'Unified Model Output (Vn13.2):'
  time            '09:20'
  title           'Stash code = 1755'
  valid_max       0.063627265
  valid_min       -0.0002616638

In [ ]: #print(planet_transm_test.coord("t")[0])

In [ ]: #planet_transm = planet_transm_test.extract(iris.Constraint(t = 2004-05-13 09:20:00))
#print(planet_transm)

In [ ]: planet_transm = planet_transm_test.collapsed('t', iris.analysis.MEAN)
print(planet_transm)

Stash code = 1755 / (unknown) (pseudo: 260; latitude: 90; longitude: 144)
Dimension coordinates:
  pseudo           x                -                -                -
  latitude         -                x                -                -
  longitude        -                -                x                -
Scalar coordinates:
  t                2004-05-19 08:20:00, bound=(2004-05-13 09:20:00, 2004-05-25 07:20:00)
Cell methods:
  0
Attributes:
  0
  t: mean
Attributes:
  date             '13/05/04'
  history          'Sun Jul 23 18:39:15 BST 2023 - XCONV V1.93 13-October-2015'
  name             'unspecified'
  source          'Unified Model Output (Vn13.2):'
  time            '09:20'
  title           'Stash code = 1755'
  valid_max       0.063627265
  valid_min       -0.0002616638

In [ ]: spectral_file = TEST_DIR / "spectral_files" / "SW_second_call" / "sp_sw_260_ultimatendwarf4"

# To keep the units of stellar and planetary parameters, store the parameters as iris cubes
stellar_constant_at_1_au = iris.cube.Cube(
    300.85538168664425, units="W m^-2", long_name="stellar_constant_at_1_au"
)
stellar_radius = iris.cube.Cube(695709090.0, units="m", long_name="stellar_radius")
planet_radius = iris.cube.Cube(71660909.0, units="m", long_name="planet_radius")
planet_domain_height = iris.cube.Cube(1.8e6, units="m", long_name="height_domain")
planet_top_of_atmosphere = planet_radius + planet_domain_height

In [ ]: spectral_bands = read_spectral_bands(spectral_file)
spectral_bands[0:10]

Out[ ]: array([(1, 1.75e-07, 1.80e-07), (2, 1.80e-07, 1.85e-07),
(3, 1.85e-07, 1.90e-07), (4, 1.90e-07, 1.95e-07),
(5, 1.95e-07, 2.00e-07), (6, 2.00e-07, 2.05e-07),
(7, 2.05e-07, 2.10e-07), (8, 2.10e-07, 2.15e-07),
(9, 2.15e-07, 2.20e-07), (10, 2.20e-07, 2.25e-07)])
dtype=('spectral_band_index', 'c4'), ('lower_wavelength_limit', 'f4'), ('upper_wavelength_limit', 'f4'])

In [ ]: normalized_stellar_flux = read_normalized_stellar_flux(spectral_file)

plt.plot(
    normalized_stellar_flux.coord("spectral_band_index").points,
    normalized_stellar_flux.data,
)
plt.xlabel("Spectral band index")
plt.ylabel("Normalized stellar flux");



In [ ]: stellar_flux = calc_stellar_flux(spectral_file, stellar_constant_at_1_au)

plt.plot(stellar_flux.coord("spectral_band_index").points, stellar_flux.data)
plt.xlabel("Spectral band index")
plt.ylabel("Stellar flux [W m^-2]");

(260,)



In [ ]: print(stellar_flux)

stellar_flux / (kg.s-3) (spectral_band_index: 260)
Dimension coordinates:
  spectral_band_index x

In [ ]: qplt.contourf(planet_transm[100, ...]);

Stash code = 1755



In [ ]: print(type(stellar_flux))
print(type(planet_transm))

<class 'iris.cube.Cube'>
<class 'iris.cube.Cube'>

In [ ]: print(stellar_flux)
print("\n", stellar_flux.shape)
print(planet_transm.shape)

stellar_flux / (kg.s-3) (spectral_band_index: 260)
Dimension coordinates:
  spectral_band_index x

(260,)
(260, 90, 144)

In [ ]: print(stellar_flux.name())
print(stellar_flux.units)
print(stellar_flux.attributes)

stellar_flux
kg.s-3
{}

In [ ]: print(planet_transm.name())
print(planet_transm.units)
print(planet_transm.attributes)

Stash code = 1755
unknown
('history': 'Sun Jul 23 18:39:15 BST 2023 - XCONV V1.93 13-October-2015', 'date': '13/05/04', 'name': 'unspecified', 'source': 'Unified Model Output (Vn13.2):', 'time': '09:20', 'title': 'Stash code = 1755', 'valid_max': 0.063627265, 'valid_min': -0.0002616638)

In [ ]: planet_transm_core_data()

Out[ ]: array([[1.5331866e-12, 3.68998492e-12, 6.47228513e-12, 2.33542650e-11,
5.48528357e-11, 1.02893258e-10, 1.37700779e-09, 5.36254629e-09,
1.17253574e-08, 3.88362279e-08, 6.01128535e-08, 1.71683816e-07,
6.2848692e-07, 1.08929170e-06, 1.01849522e-06, 2.09403299e-06,
6.4722943e-06, 2.26645905e-05, 3.19888350e-05, 2.63679907e-05,
5.1906994e-05, 4.48253149e-05, 1.05097804e-04, 3.03318429e-04,
2.40698879e-04, 2.46978621e-04, 6.40566049e-04, 3.13863954e-03,
5.53335630e-03, 4.69536348e-03, 1.08395607e-02, 1.13293655e-02,
5.90748345e-03, 5.40210946e-03, 4.10911952e-03, 9.89801932e-03,
6.8244568e-03, 8.06181569e-03, 1.52972718e-02, 4.74316273e-03,
1.53890221e-02, 1.42420712e-02, 1.01217242e-02, 1.36763125e-02,
2.51870255e-02, 4.08202088e-02, 6.01253201e-02, 5.46166879e-02,
4.37095759e-02, 1.18647965e-02, 2.74728705e-02, 5.55366671e-02,
7.15450841e-02, 6.72208636e-02, 7.26068246e-02, 8.30274012e-02,
8.36321636e-02, 7.63281519e-02, 8.52857250e-02, 1.10468974e-01,
7.63954550e-02, 6.20265199e-02, 9.34317988e-02, 1.69859372e-01,
9.31910690e-02, 7.54305085e-02, 8.54124320e-02, 1.00579935e-01,
7.55667351e-02, 1.30969036e-01, 1.50719588e-01, 1.70985065e-01,
1.73628693e-01, 1.69722412e-01, 1.20456092e-01, 1.39019089e-01,
1.54816955e-01, 1.14440392e-01, 1.41964693e-01, 1.77588771e-01,
1.85407606e-01, 1.95692418e-01, 8.8526269e-02, 9.31544835e-02,
1.30998248e-01, 1.78813567e-01, 2.36739842e-01, 2.57726232e-01,
1.42776176e-01, 1.04885534e-01, 1.33567025e-01, 1.88187161e-01,
1.71193637e-01, 2.23604234e-01, 2.40844693e-01, 2.47904358e-01,
2.08457072e-01, 2.84468920e-01, 2.09146695e-01, 1.71133693e-01,
1.76101260e-01, 1.99360850e-01, 1.95045872e-01, 2.51066509e-01,
3.38629526e-01, 8.10820604e-01, 5.06891406e-01, 8.04480527e-01,
1.33345300e+00, 1.36745115e+00, 1.50993806e+00, 1.16341802e+00,
1.09885999e+00, 1.33050975e+00, 1.52606485e+00, 1.81060079e+00,
2.08159788e+00, 2.11025746e+00, 1.97812063e+00, 1.54066952e+00,
1.43156937e+00, 1.61611888e+00, 1.89987676e+00, 2.02641513e+00,
2.06943350e+00, 2.27122101e+00, 2.51867615e+00, 2.22321257e+00,
2.08930292e+00, 2.23698740e+00, 2.38848209e+00, 2.43312699e+00,
2.41659866e+00, 2.50748192e+00, 2.33753719e+00, 2.37675096e+00,
2.42966644e+00, 2.40304599e+00, 2.37898350e+00, 2.46182920e+00,
2.49293624e+00, 2.49634927e+00, 2.52971442e+00, 2.53230991e+00,
2.53139926e+00, 2.42216526e+00, 2.42121008e+00, 2.43337216e+00,
2.37339452e+00, 2.35199867e+00, 2.43866575e+00, 2.36404419e+00,
2.32578340e+00, 2.33104763e+00, 2.31757271e+00, 4.70917475e+00,
4.70961904e+00, 4.51323602e+00, 4.80700404e+00, 4.41310720e+00,
4.32640646e+00, 4.05514170e+00, 3.64391052e+00, 3.58907330e+00,
3.47069730e+00, 3.26579820e+00, 3.13764670e+00, 3.15363762e+00,
3.02089070e+00, 3.02342720e+00, 2.95900440e+00, 3.02507615e+00,
2.99759454e+00, 3.02873748e+00, 3.07017164e+00, 3.03408946e+00,
3.05230063e+00, 3.09454174e+00, 3.03501157e+00, 3.00336569e+00,
2.80941264e+00, 2.62092276e+00, 2.44434250e+00, 2.25747910e+00,
2.12107819e+00, 1.98586182e+00, 1.91444612e+00, 1.76427436e+00,
1.70535431e+00, 1.72101951e+00, 1.71008902e+00, 1.62707010e+00,
1.54399835e+00, 1.53008423e+00, 1.45969110e+00, 1.38129695e+00,
1.45996529e+00, 1.43281397e+00, 1.40108108e+00, 1.39109429e+00,
1.39053344e+00, 1.39154932e+00, 1.38648115e+00, 1.38456494e+00,
1.37791316e+00, 1.34492094e+00, 1.35510499e+00, 1.35073999e+00,
1.32108558e+00, 1.25537435e+00, 1.21416056e+00, 1.11107159e+00,
1.08785850e+00, 1.06130090e+00, 9.90193230e-01, 2.22899881e+00,
1.98532640e+00, 1.88249361e+00, 1.59018188e+00, 1.56245399e+00,
1.51630432e+00, 1.38212180e+00, 1.28105491e+00, 1.19446996e+00,
1.13716939e+00, 1.06238380e+00, 1.02852997e+00, 9.73512875e-01,
9.37023030e-01, 9.10130409e-01, 8.84083576e-01, 8.61981621e-01,
8.46629041e-01, 1.63088857e+00, 1.55284652e+00, 1.47298263e+00,
1.38881688e+00, 1.31380966e+00, 1.22919223e+00, 1.13425094e+00,
1.01551352e+00, 9.21808295e-01, 8.30935490e-01, 7.14778069e-01,
6.20244000e-01, 5.72144994e-01, 5.37937124e-01, 5.03139590e-01,
4.64094325e-01, 4.32342913e-01, 1.75799585e+00, 1.29837339e+00,
1.00021986e+00, 6.82194878e-01, 5.34810973e-01, 4.23943660e-01,
3.34712496e-01, 2.73339860e-01, 2.08896070e-01, 1.36208321e+00]])

In [ ]: planet_transm_core_data()

Out[ ]:


| Array      | Chunk                         |
|------------|-------------------------------|
| Bytes      | 12.85 MiB 12.85 MiB           |
| Shape      | (260, 90, 144) (260, 90, 144) |
| Dask graph | 1 chunks in 5 graph layers    |
| Data type  | float32 numpy.ndarray 144     |




In [ ]: rp_eff_over_rs = calc_transmission_spectrum(
    planet_transm,
    spectral_file,
    stellar_constant_at_1_au,
    stellar_radius,
    planet_top_of_atmosphere,
    model=um,
)

(260,)

ValueError: Traceback (most recent call last):
~/home/soumik/Documents/Dlibya/Thesis_work_Jayesh/Unified_Model_GCM/Learning Iris/aeolus_my_spectra.ipynb Cell 23 in <cell line: 1>()
----> ca href="vscode-notebook-cell:/home/soumik/Documents/Dlibya/Thesis_work_Jayesh/Unified_Model_GCM/Learning20Iris/aeolus_my_spectra.ipynbX23sZmsZQ3Q3D?line=0"1</ca> rp_eff_over_rs = calc_transmission_spectrum(
<ca href="vscode-notebook-cell:/home/soumik/Documents/Dlibya/Thesis_work_Jayesh/Unified_Model_GCM/Learning20Iris/aeolus_my_spectra.ipynbX23sZmsZQ3Q3D?line=1"2</ca> planet_transm,
<ca href="vscode-notebook-cell:/home/soumik/Documents/Dlibya/Thesis_work_Jayesh/Unified_Model_GCM/Learning20Iris/aeolus_my_spectra.ipynbX23sZmsZQ3Q3D?line=2"3</ca> spectral_file,
<ca href="vscode-notebook-cell:/home/soumik/Documents/Dlibya/Thesis_work_Jayesh/Unified_Model_GCM/Learning20Iris/aeolus_my_spectra.ipynbX23sZmsZQ3Q3D?line=3"4</ca> stellar_constant_at_1_au,
<ca href="vscode-notebook-cell:/home/soumik/Documents/Dlibya/Thesis_work_Jayesh/Unified_Model_GCM/Learning20Iris/aeolus_my_spectra.ipynbX23sZmsZQ3Q3D?line=4"5</ca> stellar_radius,
<ca href="vscode-notebook-cell:/home/soumik/Documents/Dlibya/Thesis_work_Jayesh/Unified_Model_GCM/Learning20Iris/aeolus_my_spectra.ipynbX23sZmsZQ3Q3D?line=5"6</ca> planet_top_of_atmosphere,
<ca href="vscode-notebook-cell:/home/soumik/Documents/Dlibya/Thesis_work_Jayesh/Unified_Model_GCM/Learning20Iris/aeolus_my_spectra.ipynbX23sZmsZQ3Q3D?line=6"7</ca> model=um,
<ca href="vscode-notebook-cell:/home/soumik/Documents/Dlibya/Thesis_work_Jayesh/Unified_Model_GCM/Learning20Iris/aeolus_my_spectra.ipynbX23sZmsZQ3Q3D?line=7"8</ca> )
File ~/anaconda3/lib/python3.9/site-packages/aeolus/synthobs.py:172, in calc_transmission_spectrum(trans_flux, spectral_file, stellar_constant_at_1_au, stellar_radius, planet_top_of_atmosphere, model)
170 coord_rename("spectral_band_index")
171 break
--> 172 flux_ratio = trans_flux / stellar_flux
173 # Calculate the ratio of the effective planetary radius to the stellar radius
174 rp_eff_over_rs_squared = (planet_top_of_atmosphere / stellar_radius) ** 2 - flux_ratio
File ~/anaconda3/lib/python3.9/site-packages/iris/common/lenient.py:124, in _lenient_client.<locals>._lenient_client_outer.<locals>._lenient_client_inner(*args, **kwargs)
118 """
119 Closure wrapper function to register the wrapped function/method
120 as active at runtime before executing it.
121 """
122 """
123 with _LENIENT.context(*services, active=qualname(func)):
--> 124 result = func(*args, **kwargs)
125 return result
File ~/anaconda3/lib/python3.9/site-packages/iris/analysis/math.py:537, in _binary_op_common(operation_function, operation_name, cube, other, dim, in_place)
534 else:
535 op = operator.truediv
--> 537 result = _binary_op_common(
538 op,
539 "divide",
540 cube,
541 other,
542 new_unit,
543 new_dtype=new_dtype,
544 dim=dim,
545 in_place=in_place,
546 )
548 return result
File ~/anaconda3/lib/python3.9/site-packages/iris/analysis/math.py:897, in _binary_op_common(operation_function, operation_name, cube, other, new_unit, new_dtype, dim, in_place, sanitize_metadata)
893 rhs = broadcast_cube_coord_data(cube, other, operation_name, dim=dim)
894 elif isinstance(oper, Cube):
895 # Prepare to resolve the cube operands and associated coordinate
896 # metadata into the resultant cube.
--> 897 resolver = Resolve(cube, other)
898 # Get the broadcast, auto-transposed safe versions of the cube operands.
900 cube = resolver.lhs_cube_resolved
File ~/anaconda3/lib/python3.9/site-packages/iris/common/resolve.py:327, in Resolve._init_(self, lhs, rhs)
323 self._broadcast_shape = None # if set in _as_compatible_cubes
325 if lhs is not None or rhs is not None:
326 # Attempt to resolve the cube operands.
--> 327 self(lhs, rhs)
File ~/anaconda3/lib/python3.9/site-packages/iris/common/resolve.py:387, in Resolve._call_(self, lhs, rhs)
384 self._debug_items(self.category_common, title="common")
385 logger.debug(f"map_rhs_to_lhs={self.map_rhs_to_lhs}")
--> 387 self._metadata_mapping()
388 self._metadata_prepare()
390 return self
File ~/anaconda3/lib/python3.9/site-packages/iris/common/resolve.py:1234, in Resolve._metadata_mapping(self)
1228 # Mapping common aux coordinates, mapping={self.mapping}
1229 )
1231 if not self.mapped:
1232 # Attempt to complete the mapping using src/tgt free dimensions.
1233 # Note that, this may not be possible and result in an exception.
--> 1234 self._free_mapping(
1235 src_dim_coverage,
1236 tgt_dim_coverage,
1237 src_aux_coverage,
1238 tgt_aux_coverage,
1239 )
1241 # Attempt to transpose/reshape the cubes into compatible broadcast shapes.
1242 # Note that, this may not be possible and result in an exception.
1243 self._as_compatible_cubes()
File ~/anaconda3/lib/python3.9/site-packages/iris/common/resolve.py:1104, in Resolve._free_mapping(self, src_dim_coverage, tgt_dim_coverage, src_aux_coverage, tgt_aux_coverage)
1097 plural = "s" if len(src_undef) > 1 else ""
1098 msg = (
1099 "Insufficient matching coordinate metadata to resolve cubes, "
1100 "cannot map dimension {plural} {tuple(sorted(src_undef))} = "
1101 f"to the {self._src_cube.position} cube (src_cube.shape, src_cube.name{1:r} "
1102 f"to the {self._tgt_cube.position} cube (tgt_cube.shape, tgt_cube.name{1:r} "
1103 )
1104 )
--> 1104 raise ValueError(msg)
1106 # Update the mapping.
1107 self.mapping.update(free_mapping)
ValueError: Insufficient matching coordinate metadata to resolve cubes, cannot map dimension (0,) of the RHS cube ((260,), 'stellar_flux') to the LHS cube ((260,), 'Stash code = 1755').

```