ECSN-Profile-Comparison

May 9, 2019

1 ECSN Profile Comparison

In [1]:
```
from StarKiller.initialization import starkiller_initialize
from StarKiller.interfaces import EosType
from StarKiller.interfaces import BurnType
from StarKiller.eos import Eos
from StarKiller.network import Network
from StarKiller.models import AmrexAstroModel
from mesa_reader import MesaData
import numpy as np
import matplotlib.pyplot as plt
import matplotlib
%matplotlib inline
```

1.1 Initialize Starkiller Microphysics

In [2]:
```
probin_file = "probin_ecsn"
```

In [3]:
```
starkiller_initialize(probin_file)
```

In [4]:
```
helmholtz = Eos()
ecsn = Network()
```

In [5]:
```
def evaluate_rhs(burn_state):
    ecsn.rhs(burn_state)
    return burn_state

def evaluate_eos(burn_state):
    eos_state = burn_state.to_eos_type()
    helmholtz.evaluate(eos_state.eos_input_rt, eos_state)
    return eos_state
```

1.2 Read MESA profile

In [6]:
```
mesa_data = MesaData("../../../../profiles/ONe6040-final.data")
```

In [7]:
```
mesa_enuc = mesa_data.eps_nuc_mc2 - mesa_data.eps_nuc_neu
```

```
In [8]: shared_species = []

        for s in ecsn.short_species_names:
            if s in mesa_data.bulk_names:
                shared_species.append(s)

        print("Species shared between MESA and Starkiller networks:\n")
        for s in shared_species:
            print(s)
```

```
Species shared between MESA and Starkiller networks:

h1
he4
o16
o20
f20
ne20
mg24
si28
```

1.3 Evaluate the Starkiller equivalent of MESA (eps_nuc_mc2 - eps_nuc_neu)

```
In [9]: sk_burn_results = []
        sk_eos_results = []

        for zi in range(len(mesa_data.zone)):
            density = 10.0**mesa_data.logRho[zi]
            temperature = 10.0**mesa_data.logT[zi]

            # Set mass fractions for Starkiller network by zeroing
            # mass fractions for species missing in the MESA network and ignoring
            # species missing in the Starkiller network.
            #
            # Then renormalize the Starkiller mass fractions to sum to 1.
            mass_fractions = []

            for s in ecsn.short_species_names:
                if s in mesa_data.bulk_names:
                    mass_fractions.append(mesa_data.data(s)[zi])
                else:
                    mass_fractions.append(0.0)

            mass_fractions = np.array(mass_fractions)
            mass_fractions = mass_fractions/np.sum(mass_fractions)
```

```
        burn_state = BurnType()
        burn_state.state.rho = density
        burn_state.state.t = temperature
        burn_state.state.xn = mass_fractions

        # Evaluate the RHS and EOS in Starkiller
        sk_burn_results.append(evaluate_rhs(burn_state))
        sk_eos_results.append(evaluate_eos(burn_state))
```

```
In [10]: sk_eps = []

         for burn_state in sk_burn_results:
             sk_eps.append(burn_state.state.ydot[ecsn.net_ienuc])

         sk_eps = np.array(sk_eps)
```

1.4 Read MaestroEx initial model

```
In [11]: maestro_data = AmrexAstroModel('ECSN-ONe6040-final.hse.10240')
```

```
In [12]: maestro_data.model_data['logRho'] = np.log10(maestro_data.data('density'))
```

1.5 Evaluate energy generation rate for MaestroEx initial model

```
In [13]: sk_maestro_burn_results = []
         sk_maestro_eos_results = []

         for zi in range(maestro_data.size):
             mass_fractions = []

             for s in ecsn.short_species_names:
                 mass_fractions.append(maestro_data.data(s)[zi])

             mass_fractions = np.array(mass_fractions)
             mass_fractions = mass_fractions/np.sum(mass_fractions)

             burn_state = BurnType()
             burn_state.state.rho = maestro_data.data('density')[zi]
             burn_state.state.t = maestro_data.data('temperature')[zi]
             burn_state.state.xn = mass_fractions

             # Evaluate the RHS and EOS in Starkiller
             sk_maestro_burn_results.append(evaluate_rhs(burn_state))
             sk_maestro_eos_results.append(evaluate_eos(burn_state))
```

```
In [14]: sk_maestro_eps = []

         for burn_state in sk_maestro_burn_results:
             sk_maestro_eps.append(burn_state.state.ydot[ecsn.net_ienuc])
```

```
sk_maestro_eps = np.array(sk_maestro_eps)
```

1.6 Plot MESA, Starkiller-MESA, and Starkiller-Maestro energy generation rates

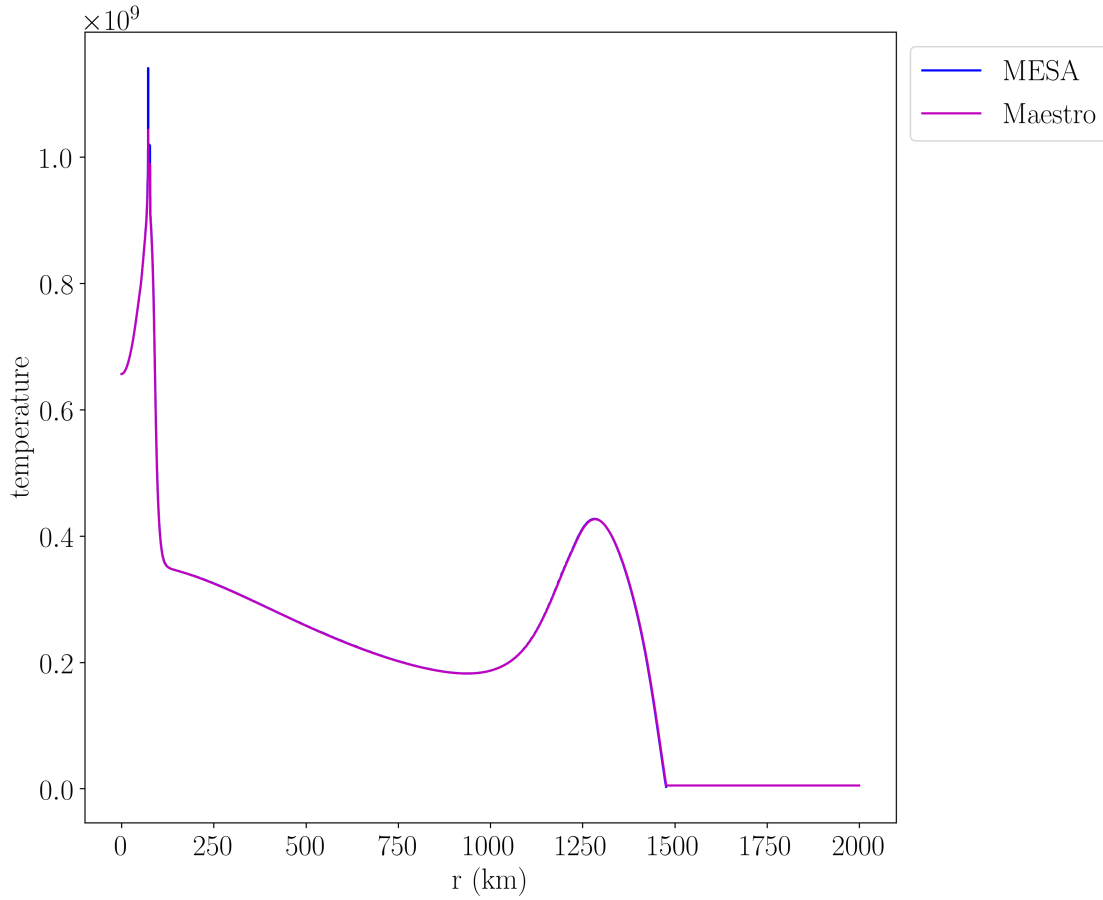
In [15]:
```
font_config = {'size': 20}
text_config = {'usetex': True}
figure_config = {'dpi': 300}

matplotlib.rc('font', **font_config)
matplotlib.rc('text', **text_config)
matplotlib.rc('figure', **figure_config)

mesa_enuc_pos = np.maximum(mesa_enuc, 0.0)
mesa_enuc_neg = np.maximum(-mesa_enuc, 0.0)

sk_enuc_pos = np.maximum(sk_eps, 0.0)
sk_enuc_neg = np.maximum(-sk_eps, 0.0)

sk_maestro_enuc_pos = np.maximum(sk_maestro_eps, 0.0)
sk_maestro_enuc_neg = np.maximum(-sk_maestro_eps, 0.0)

radius_km = mesa_data.radius_cm * 1.0e-5

# The order of zones in MESA and Maestro are reverse from each other
# so define a separate radius array for the Maestro data
maestro_radius_km = maestro_data.data('radius') * 1.0e-5

def plot_enuc(xlim=None, ylim=None):
    fig, ax = plt.subplots(facecolor='w')
    fig.set_figheight(10.0)
    fig.set_figwidth(10.0)

    ax.plot(radius_km, mesa_enuc_pos, color='b', linestyle='-',
            label=r'$\mathrm{MESA}$ $\dot{\varepsilon} > 0$')

    ax.plot(radius_km, mesa_enuc_neg, color='b', linestyle=':',
            label=r'$\mathrm{MESA}$ $\dot{\varepsilon} < 0$')

    ax.plot(radius_km, sk_enuc_pos, color='g', linestyle='-',
            label=r'$\mathrm{StarKiller-MESA}$ $\dot{\varepsilon} > 0$')

    ax.plot(radius_km, sk_enuc_neg, color='g', linestyle=':',
            label=r'$\mathrm{StarKiller-MESA}$ $\dot{\varepsilon} < 0$')

    ax.plot(maestro_radius_km, sk_maestro_enuc_pos, color='m', linestyle='-',
            label=r'$\mathrm{StarKiller-Maestro}$ $\dot{\varepsilon} > 0$')
```

```
        ax.plot(maestro_radius_km, sk_maestro_enuc_neg, color='m', linestyle=':',
                label=r'$\mathrm{StarKiller-Maestro}$ $\dot{\varepsilon} < 0$')

        ax.set_yscale('log')

        if xlim:
            ax.set_xlim(xlim)

        if ylim:
            ax.set_ylim(ylim)

        ax.set_xlabel(r'$\mathrm{r\ (km)}$')
        ax.set_ylabel(r'$\dot{\varepsilon} = \dot{\varepsilon}_{m c^2} - |\dot{\varepsilon}
        ax.legend(loc='upper left', bbox_to_anchor=(1.0, 1.0))

        plt.show()
```

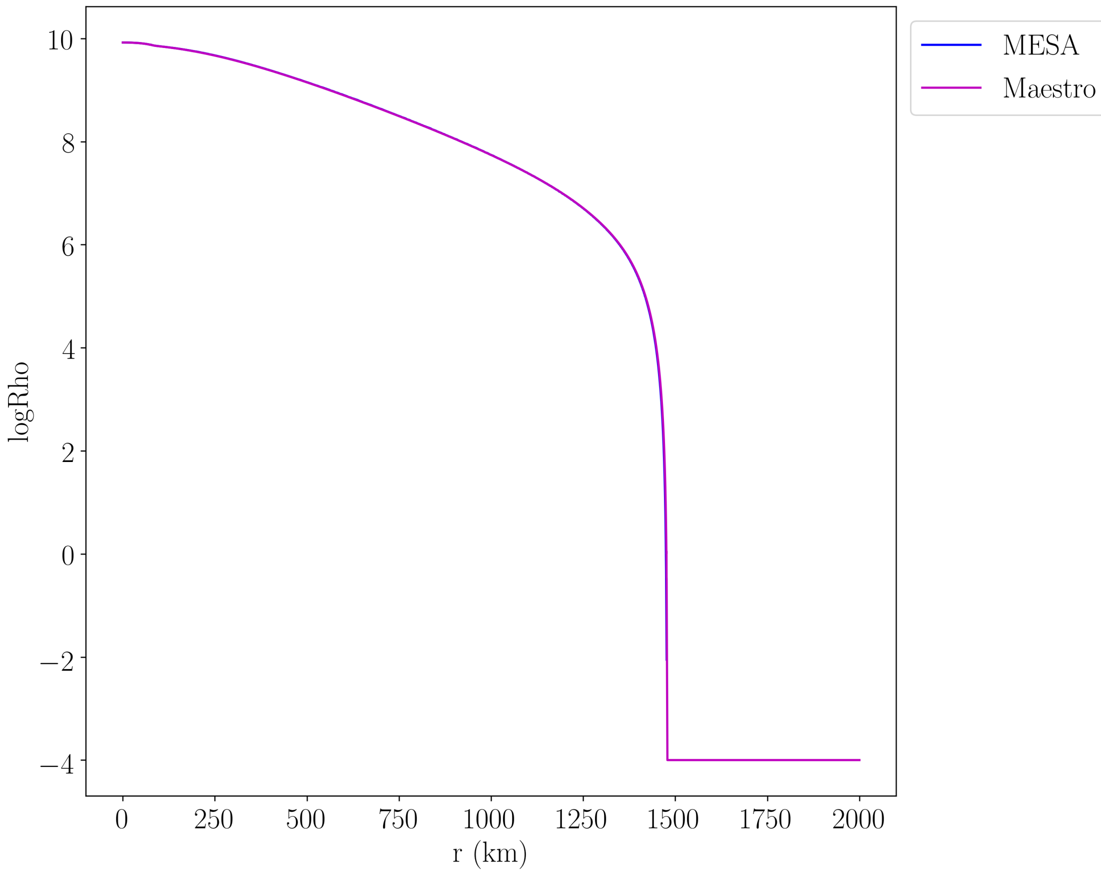
```
In [16]: def plot_field(xlim=None, ylim=None, field="temperature", logy=False):
             fig, ax = plt.subplots(facecolor='w')
             fig.set_figheight(10.0)
             fig.set_figwidth(10.0)

             ax.plot(radius_km, mesa_data.data(field), color='b',
                     label=r'$\mathrm{MESA}$')
             ax.plot(maestro_radius_km, maestro_data.data(field), color='m',
                     label=r'$\mathrm{Maestro}$')

             if xlim:
                 ax.set_xlim(xlim)

             if ylim:
                 ax.set_ylim(ylim)

             if logy:
                 ax.set_yscale('log')

             ax.set_xlabel(r'$\mathrm{r\ (km)}$')
             ax.set_ylabel(r'$\mathrm{' + '{}'.format(field) + r'}$')
             ax.legend(loc='upper left', bbox_to_anchor=(1.0, 1.0))

             plt.show()
```

1.6.1 Entire Profile

```
In [17]: plot_field(field='temperature')
```

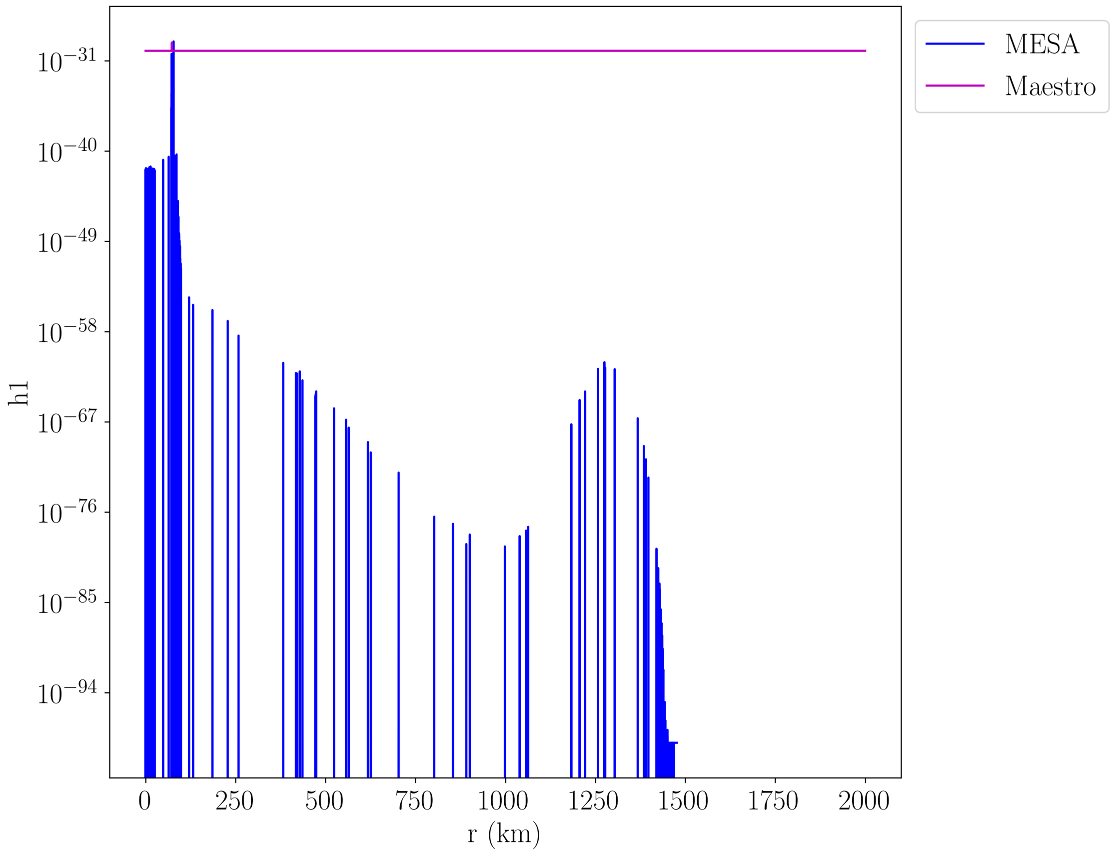


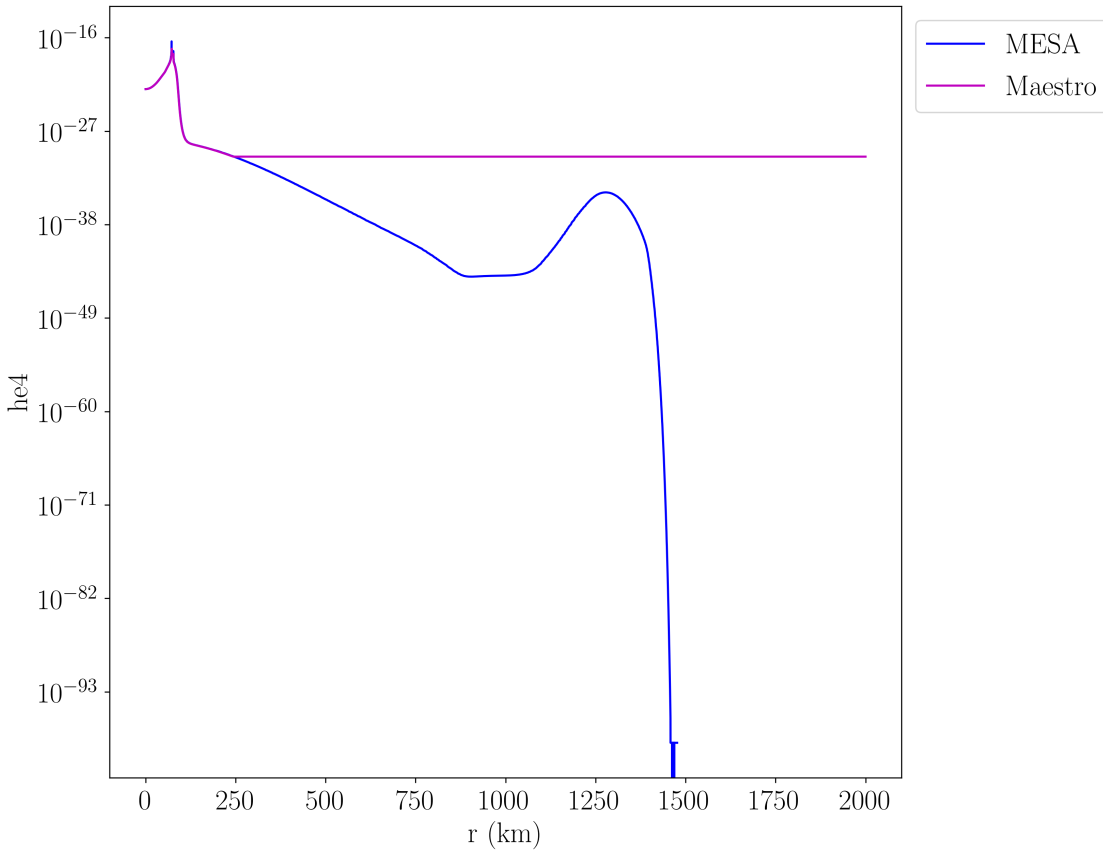

```
In [18]: plot_field(field='logRho')
```



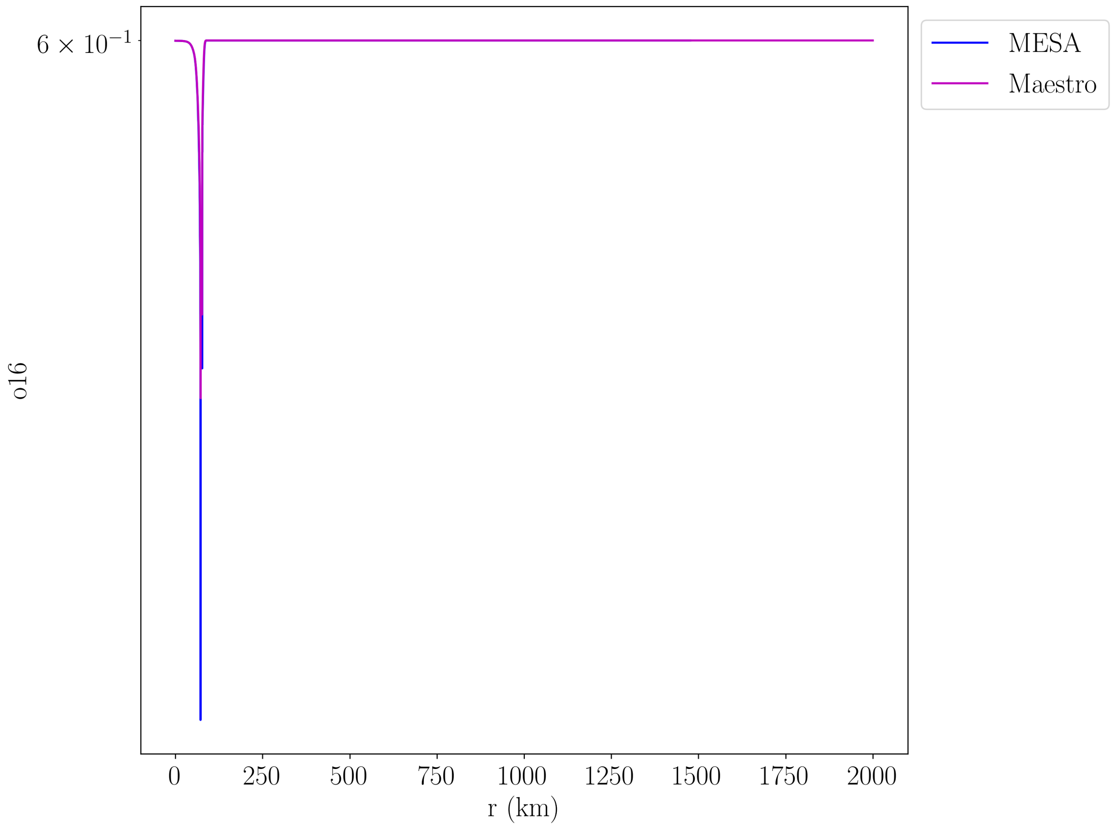
1.6.2 Plot mass fractions for species shared between MESA and Maestro

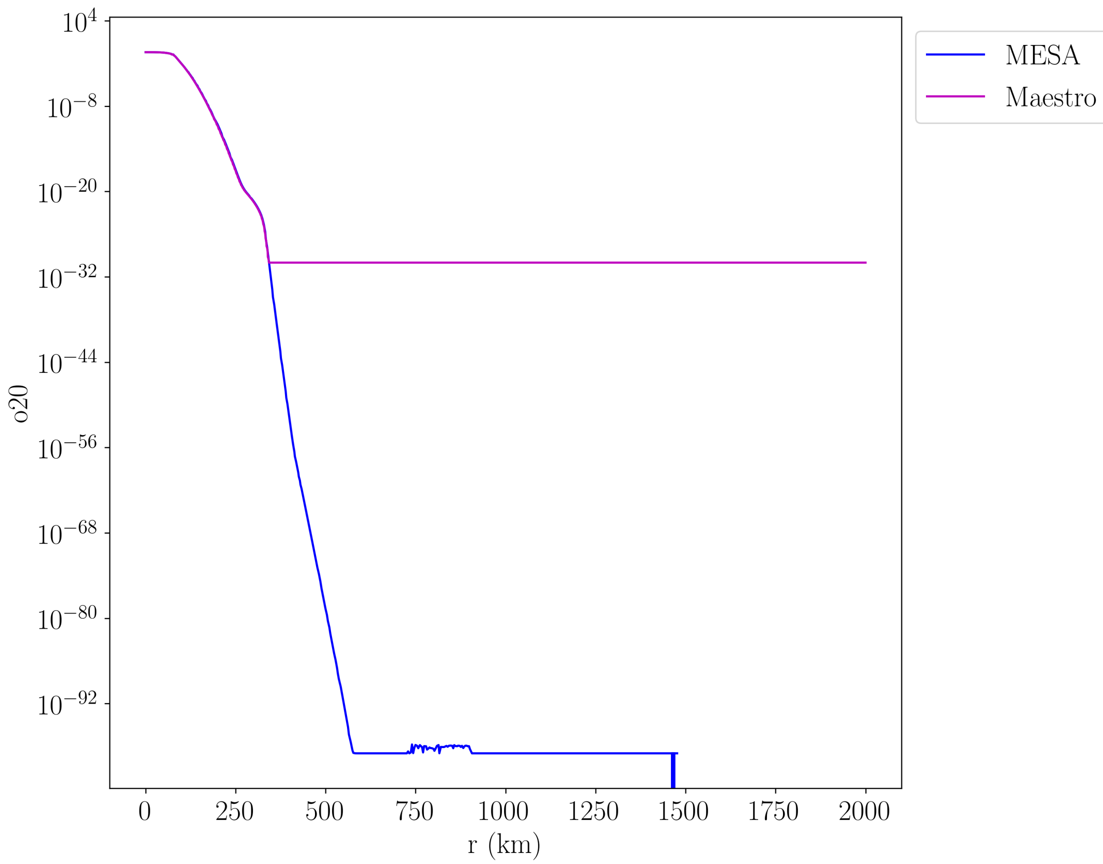
```
In [19]: for s in shared_species:
             plot_field(field=s, logy=True)
```


MESA
Maestro
h1
r (km)
10^{-31}
10^{-40}
10^{-49}
10^{-58}
10^{-67}
10^{-76}
10^{-85}
10^{-94}
0
250
500
750
1000
1250
1500
1750
2000

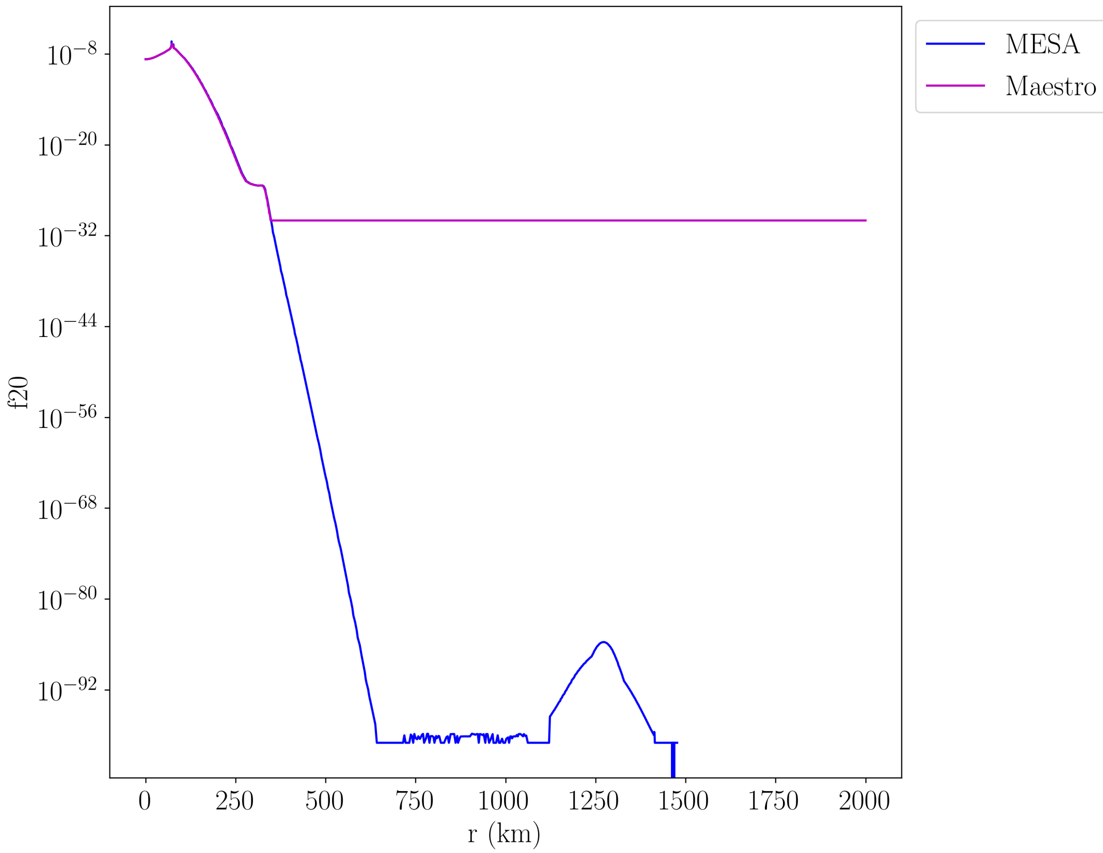


10^{-16}
10^{-27}
10^{-38}
10^{-49}
10^{-60}
10^{-71}
10^{-82}
10^{-93}
he4
0
250
500
750
1000
1250
1500
1750
2000
r (km)
MESA
Maestro

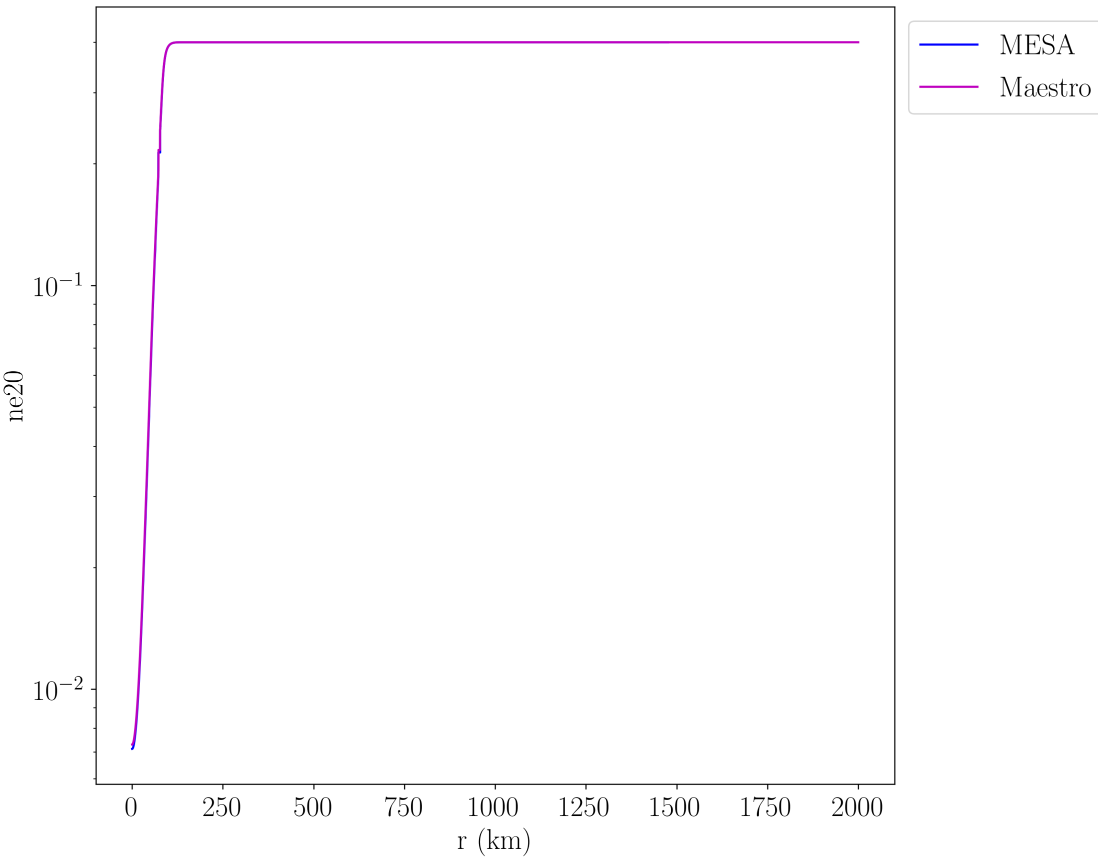

6 × 10⁻¹
o16
0
250
500
750
1000
1250
1500
1750
2000
r (km)
MESA
Maestro



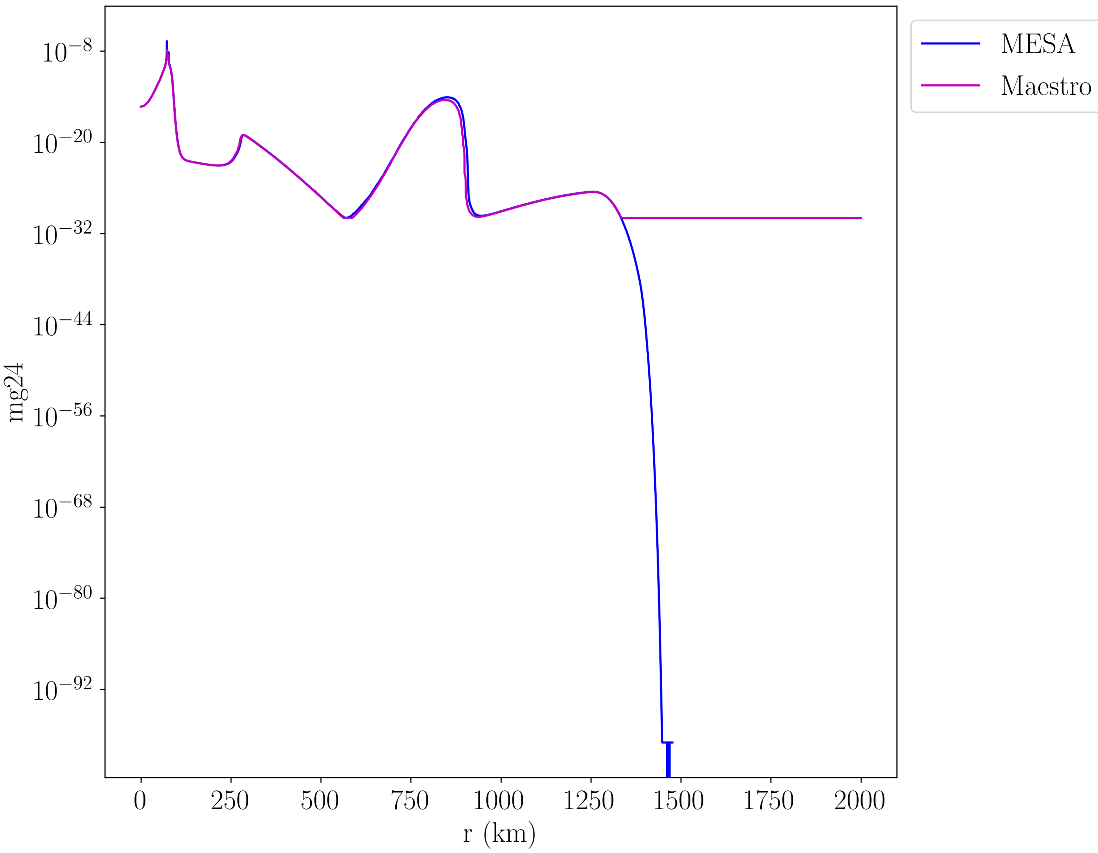
MESA
Maestro
o20
r (km)
10^{4}
10^{-8}
10^{-20}
10^{-32}
10^{-44}
10^{-56}
10^{-68}
10^{-80}
10^{-92}
0
250
500
750
1000
1250
1500
1750
2000

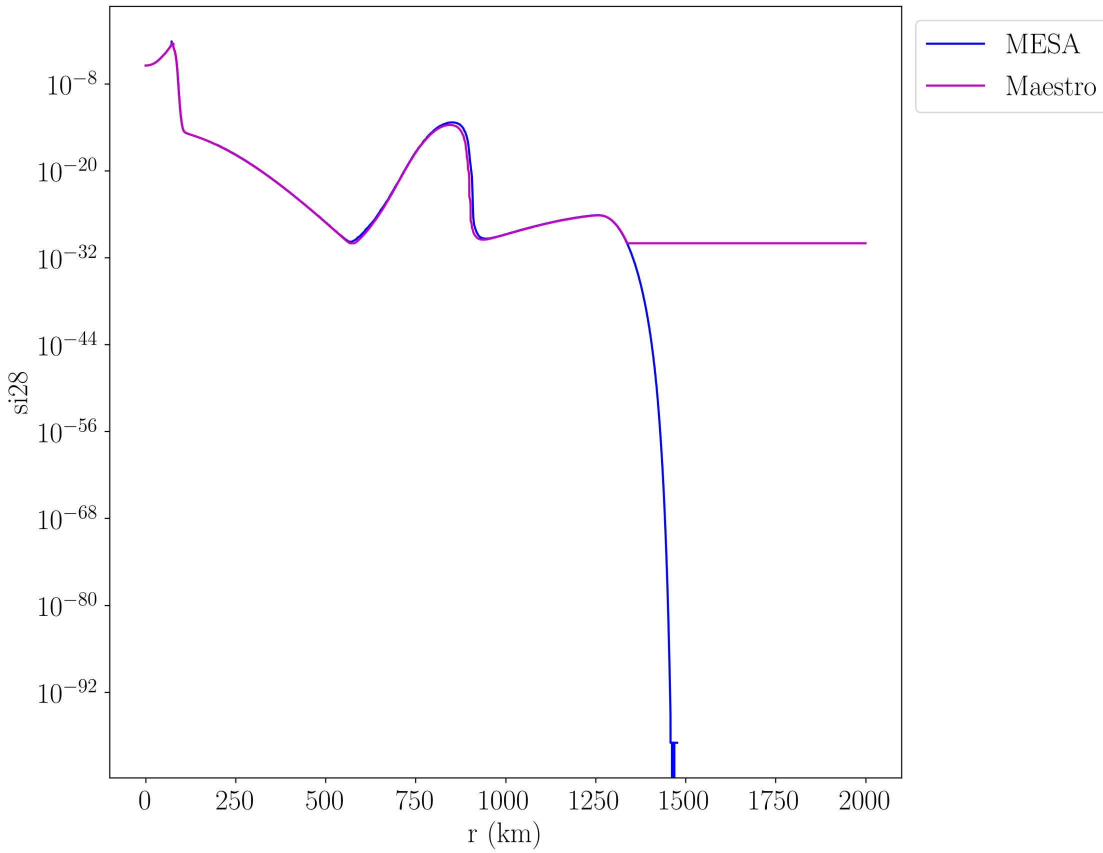


MESA
Maestro
f20
r (km)
10^{-8}
10^{-20}
10^{-32}
10^{-44}
10^{-56}
10^{-68}
10^{-80}
10^{-92}
0
250
500
750
1000
1250
1500
1750
2000



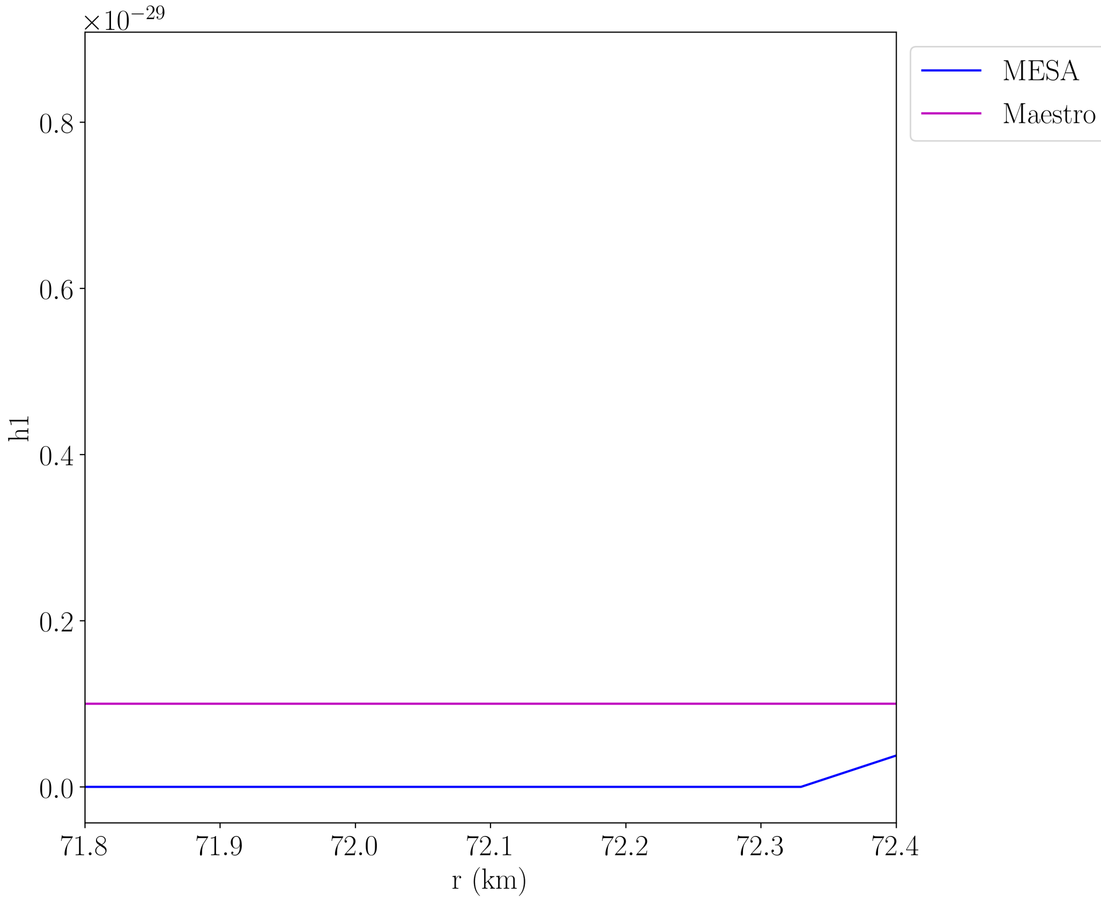
MESA
Maestro
ne20
10^{-1}
10^{-2}
0
250
500
750
1000
1250
1500
1750
2000
r (km)


10^{−8}
10^{−20}
10^{−32}
10^{−44}
10^{−56}
10^{−68}
10^{−80}
10^{−92}
mg24
0
250
500
750
1000
1250
1500
1750
2000
r (km)
MESA
Maestro

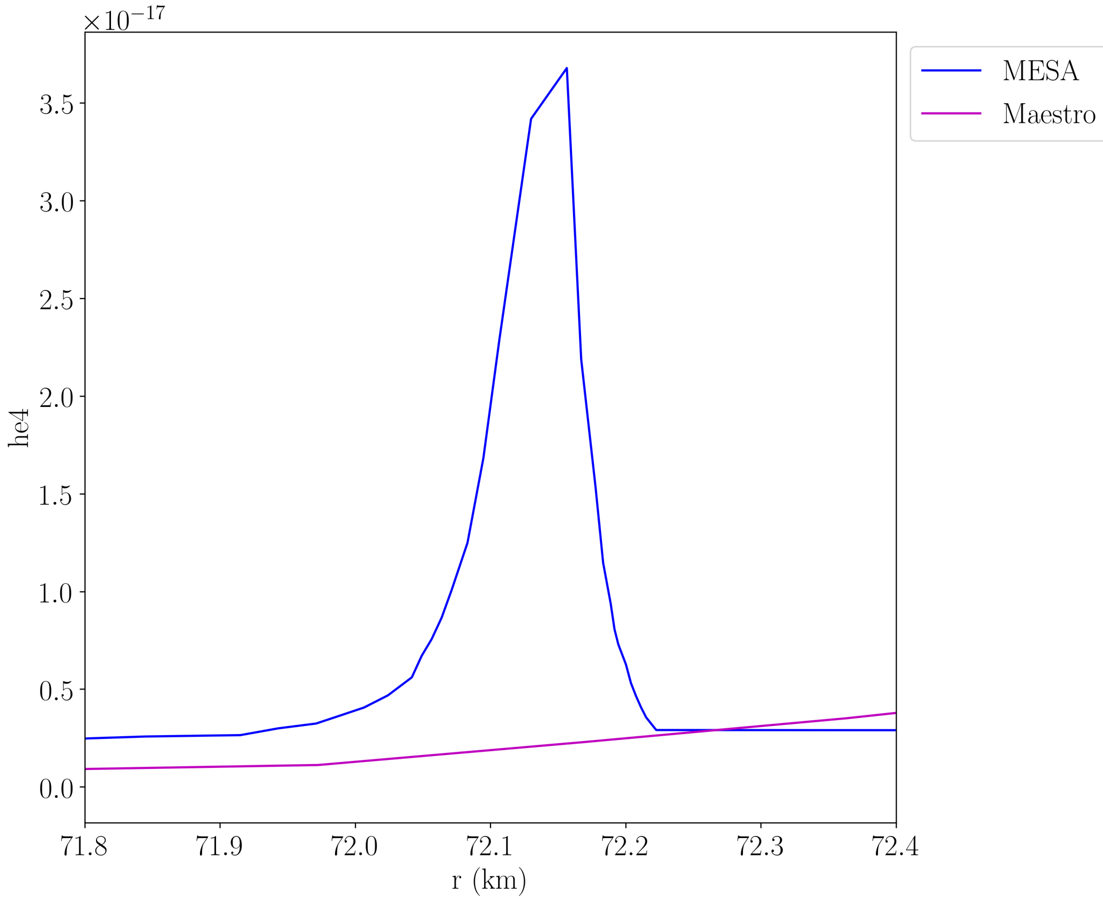


1.6.3 Zoom in on temperature peak - mass fractions for species shared between MESA and Maestro

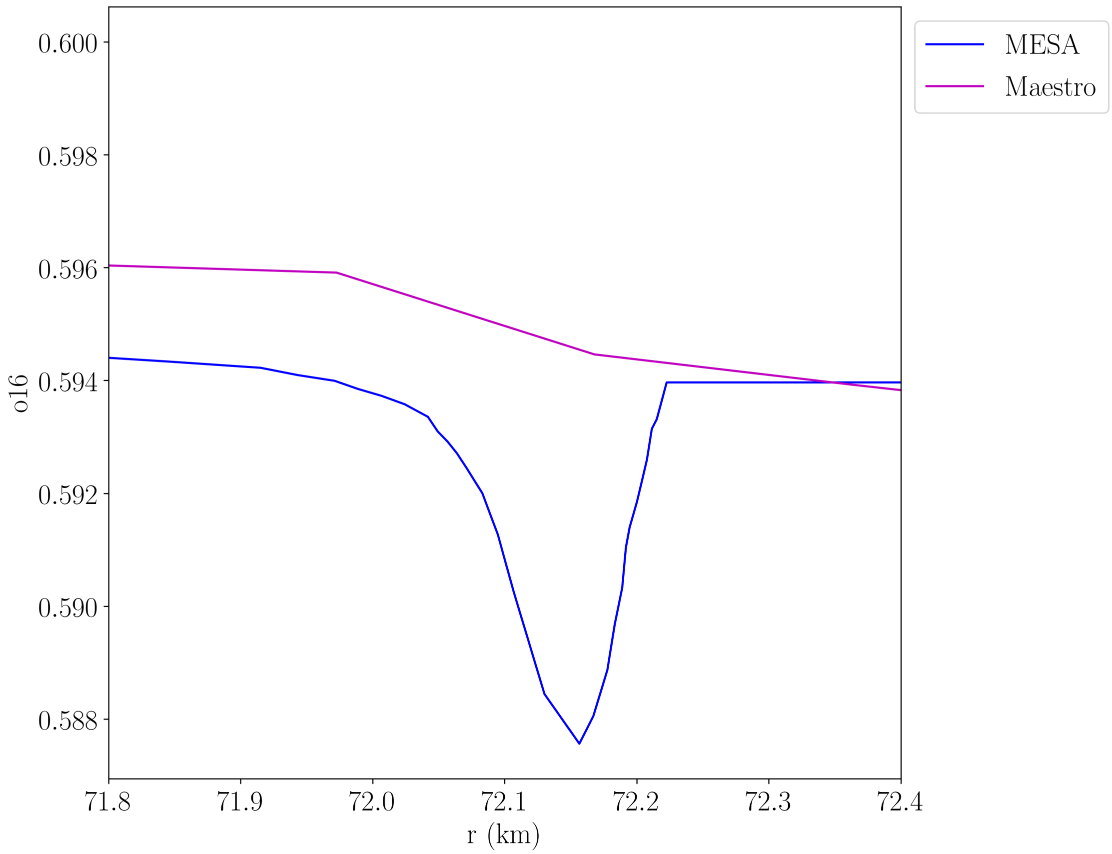
```
In [20]: for s in shared_species:
             plot_field(field=s, logy=False, xlim=[71.8, 72.4])
```



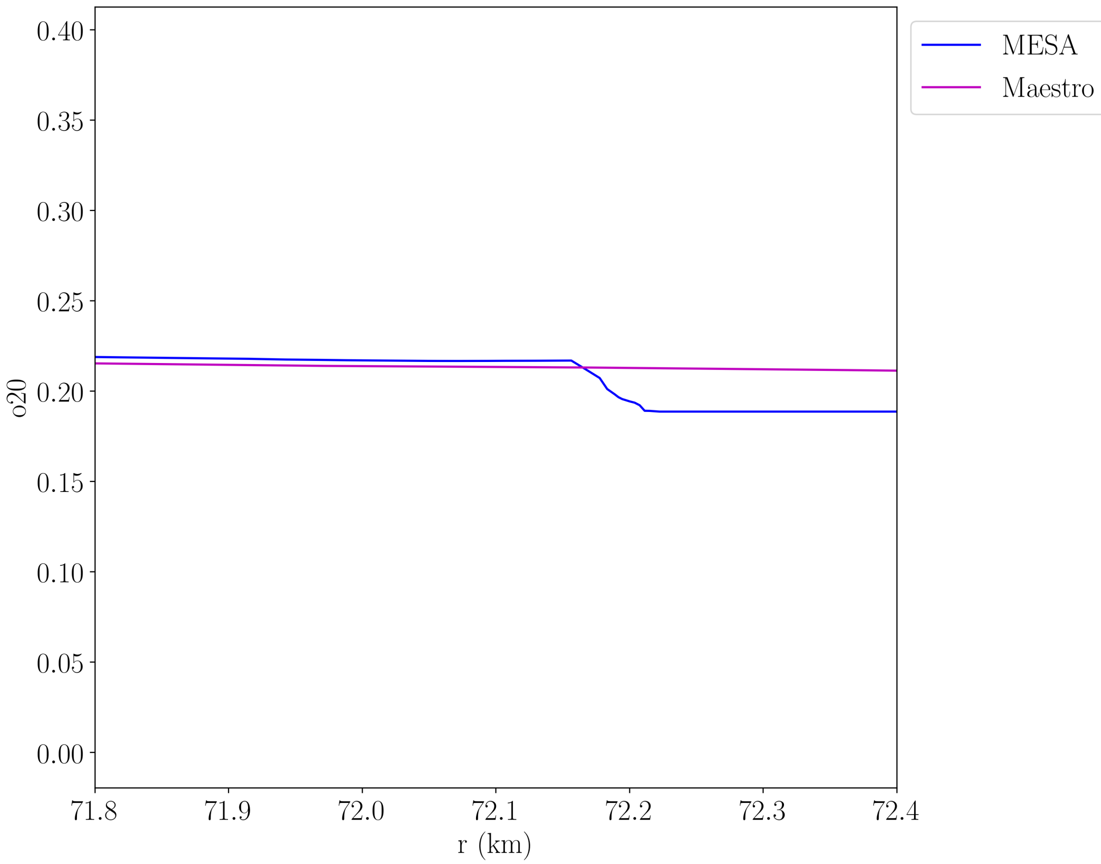
$\times 10^{-29}$
0.8
0.6
0.4
0.2
0.0
h1
71.8
71.9
72.0
72.1
72.2
72.3
72.4
r (km)
MESA
Maestro

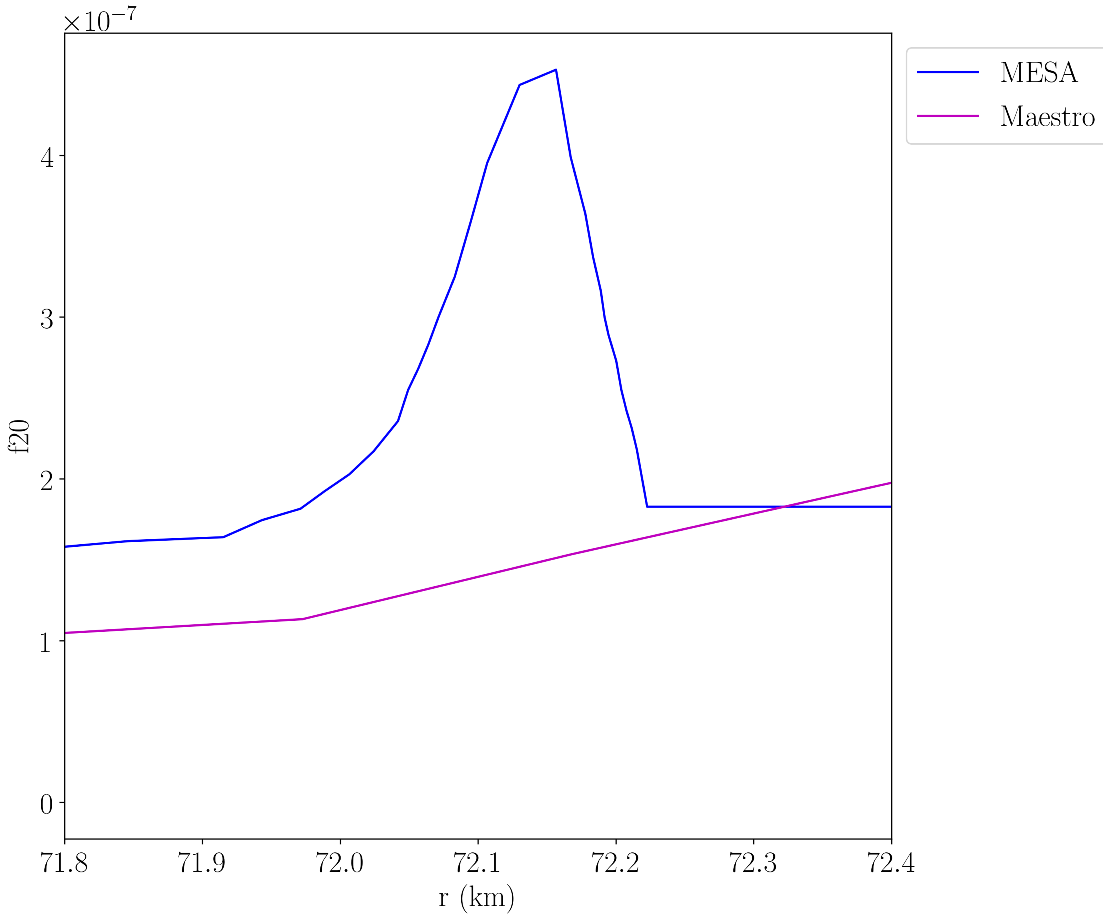


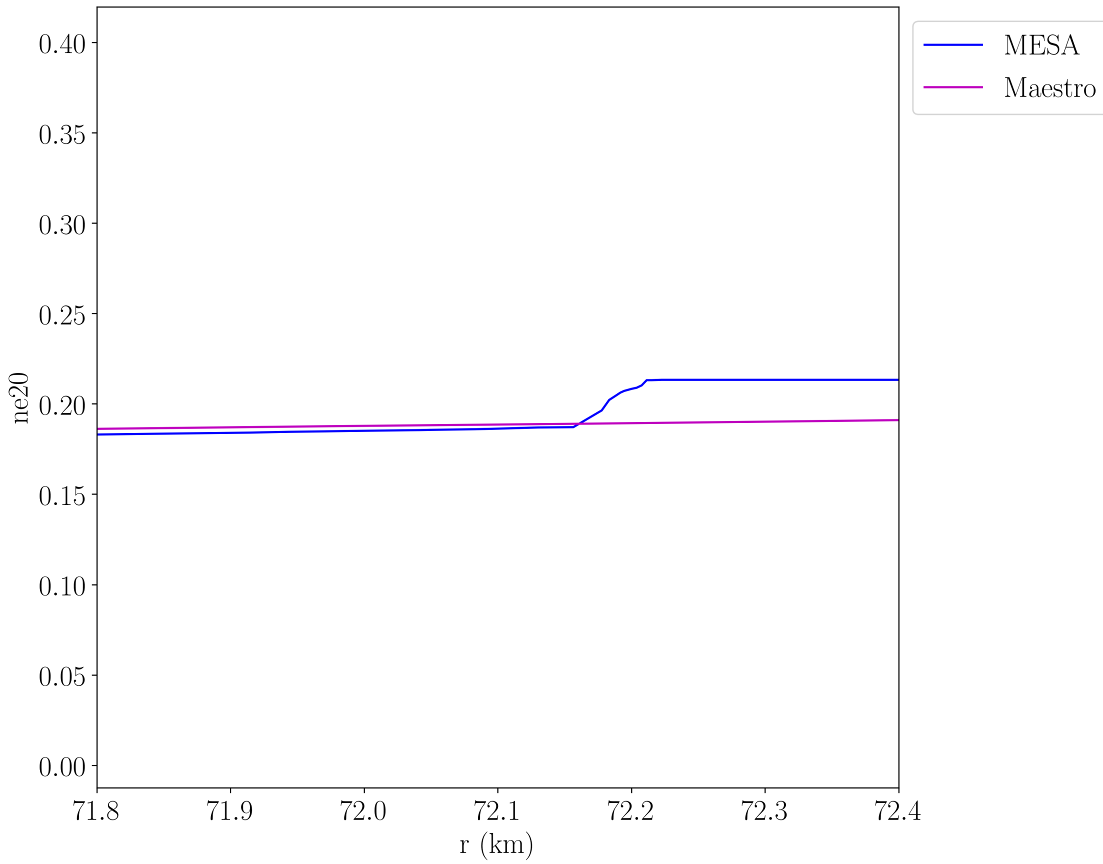
$\times 10^{-17}$
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
he4
71.8
71.9
72.0
72.1
72.2
72.3
72.4
r (km)
MESA
Maestro

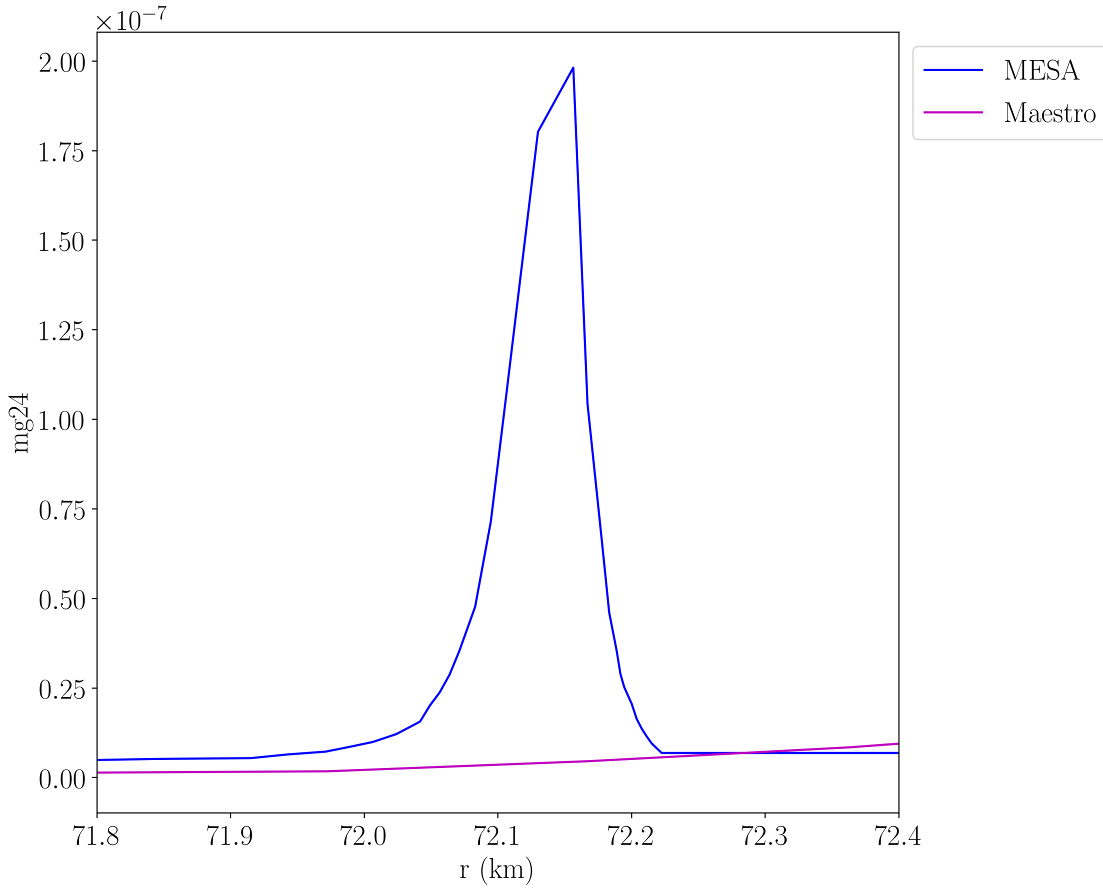


0.600
0.598
0.596
0.594
0.592
0.590
0.588
o16
71.8
71.9
72.0
72.1
72.2
72.3
72.4
r (km)
MESA
Maestro

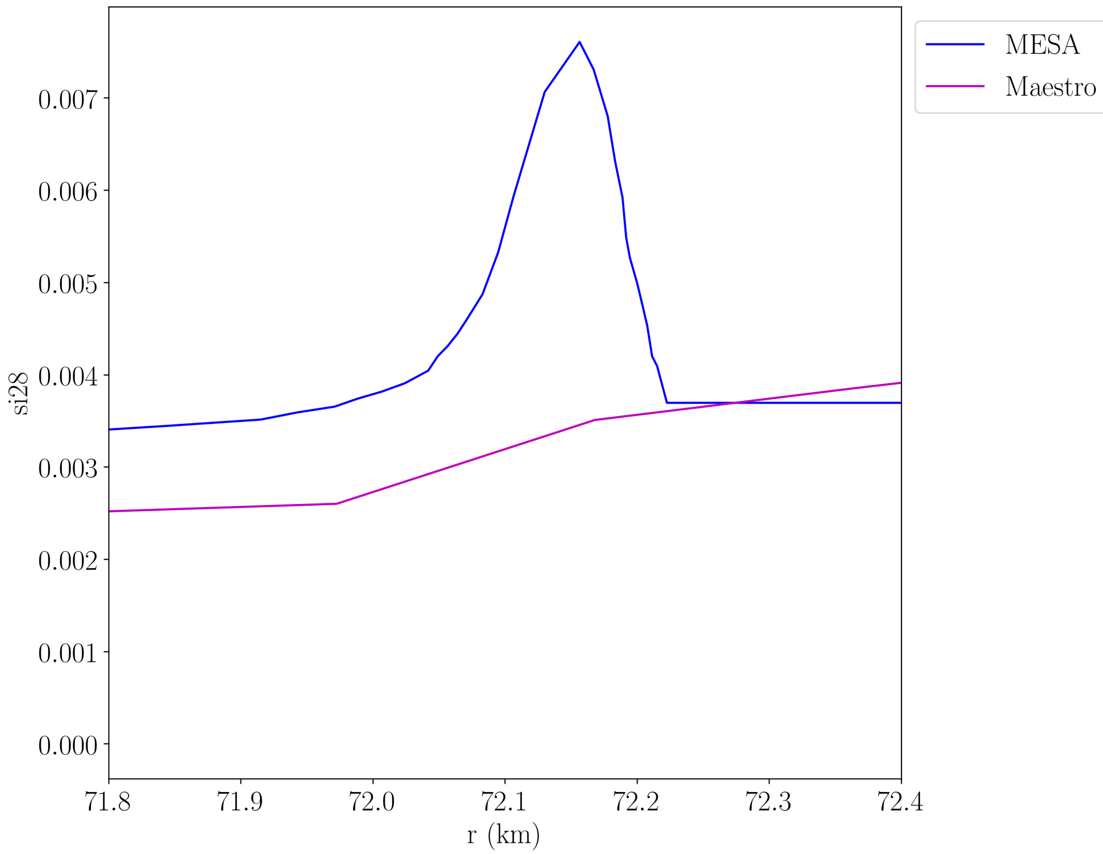

0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
o20
71.8
71.9
72.0
72.1
72.2
72.3
72.4
r (km)
MESA
Maestro


$\times 10^{-7}$
MESA
Maestro
f20
0
1
2
3
4
71.8
71.9
72.0
72.1
72.2
72.3
72.4
r (km)

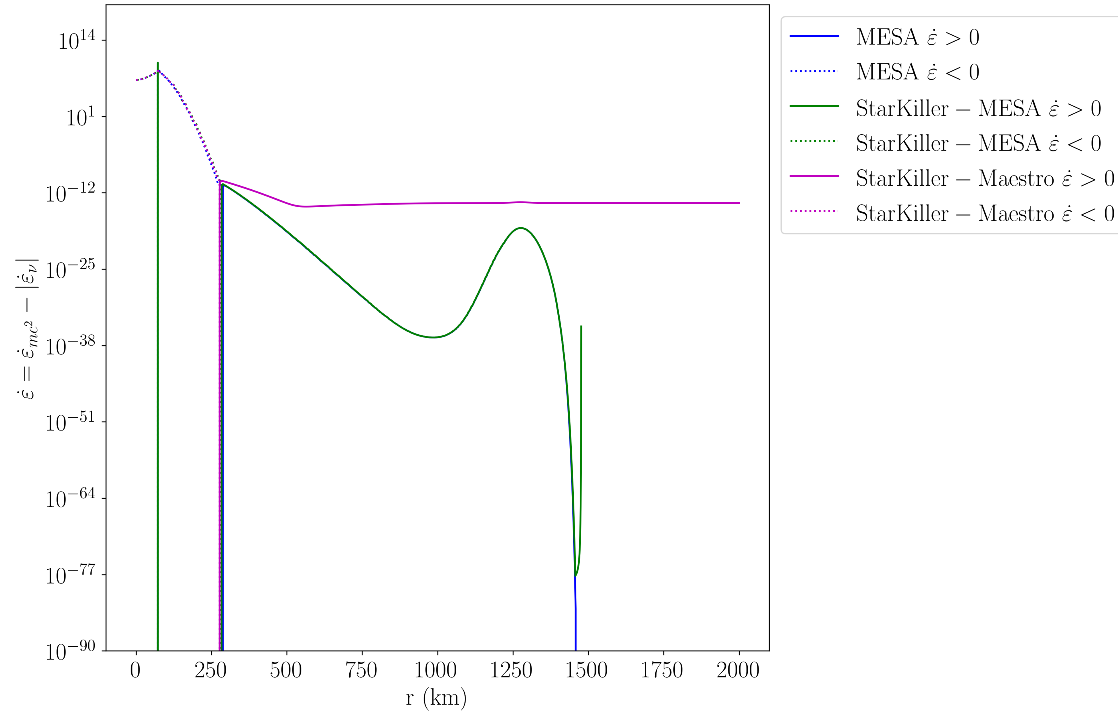

0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
ne20
71.8
71.9
72.0
72.1
72.2
72.3
72.4
r (km)
MESA
Maestro



×10^−7
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
mg24
71.8
71.9
72.0
72.1
72.2
72.3
72.4
r (km)
MESA
Maestro

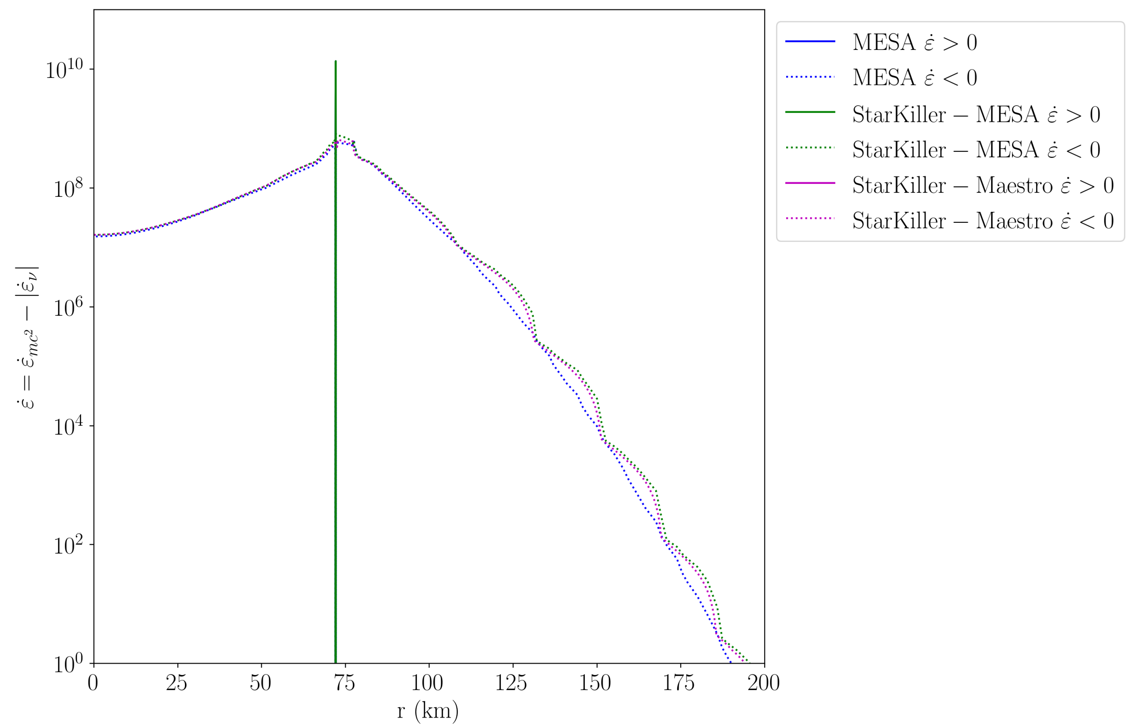


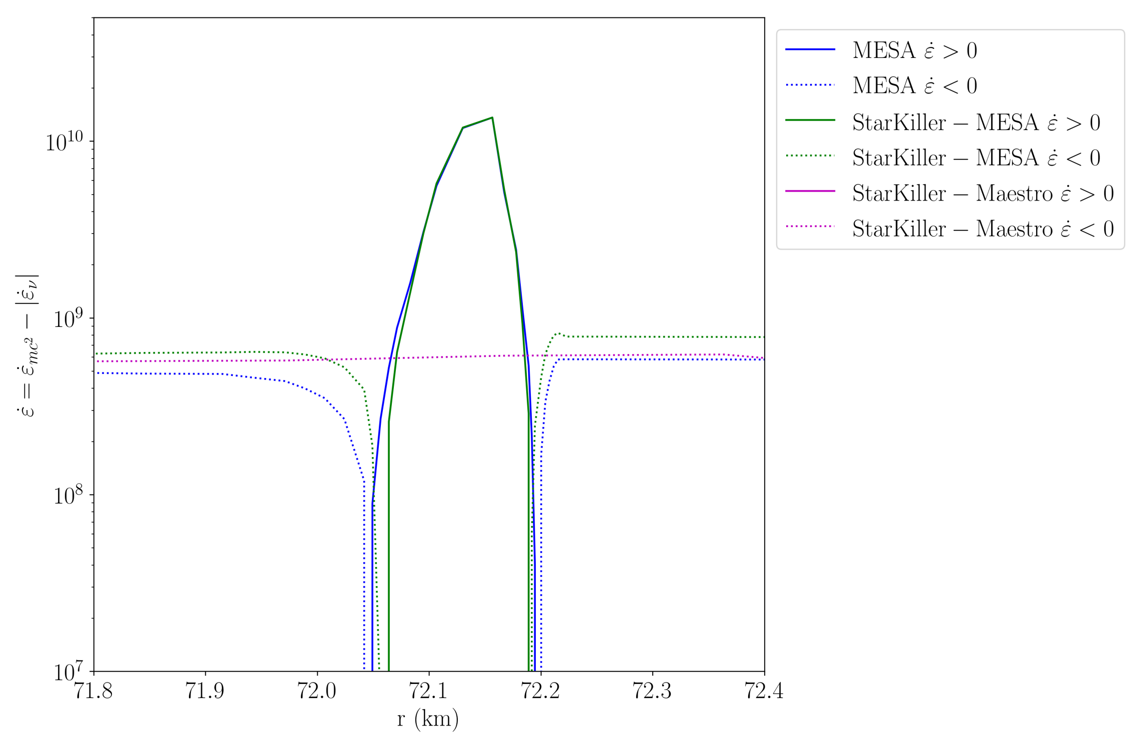

```
In [21]: plot_enuc(ylim=[1.0e-90, 1.0e20])
```



1.6.4 Zoom in on peak temperature

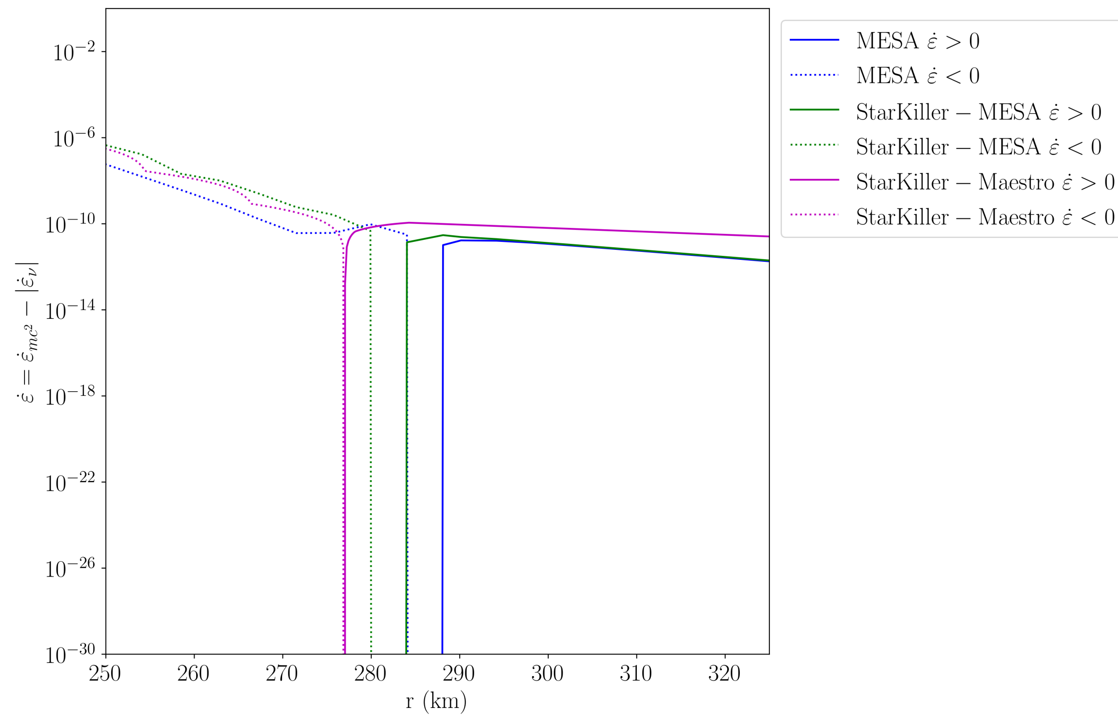
In [22]:
```
plot_enuc(xlim=[0.0, 200], ylim=[1.0, 1.0e11])
plot_enuc(xlim=[71.8, 72.4], ylim=[1.0e7, 5.0e10])
```


MESA $\dot{\varepsilon} > 0$
MESA $\dot{\varepsilon} < 0$
StarKiller − MESA $\dot{\varepsilon} > 0$
StarKiller − MESA $\dot{\varepsilon} < 0$
StarKiller − Maestro $\dot{\varepsilon} > 0$
StarKiller − Maestro $\dot{\varepsilon} < 0$
$\dot{\varepsilon} = \dot{\varepsilon}_{mc^2} - |\dot{\varepsilon}_\nu|$
r (km)


MESA $\dot{\varepsilon} > 0$
MESA $\dot{\varepsilon} < 0$
StarKiller − MESA $\dot{\varepsilon} > 0$
StarKiller − MESA $\dot{\varepsilon} < 0$
StarKiller − Maestro $\dot{\varepsilon} > 0$
StarKiller − Maestro $\dot{\varepsilon} < 0$
$\dot{\varepsilon} = \dot{\varepsilon}_{mc^2} - |\dot{\varepsilon}_\nu|$
r (km)

1.6.5 Zoom in on ~300 km radius

```
In [23]: plot_enuc(xlim=[250, 325], ylim=[1.0e-30, 1.0])
```



1.7 Print central conditions in the StarKiller-MESA model vs StarKiller-Maestro model

```
In [24]: print("StarKiller-MESA model core:")
         print("radius = {}".format(mesa_data.radius_cm[-1]))
         print(sk_burn_results[-1].state)
         sk_mesa_core = sk_burn_results[-1]
```

```
StarKiller-MESA model core:
radius = 4238.740586226362
<burn_t>{
    rho : 8357088644.807663,
    t : 656576876.0892806,
    e : 2.4376186957017733e+18,
    xn : array([0.00000000e+00, 8.96186261e-23, 5.99994533e-01, 3.92886298e-01,
       2.00272928e-09, 7.11563848e-03, 4.83382902e-16, 0.00000000e+00,
       3.52829753e-06, 0.00000000e+00, 0.00000000e+00]),
    cv : 13443541.124250965,
    cp : 13469891.007039556,
    y_e : 0.46071137008219526,
    eta : 134.03551389904297,
```

```
cs : 1086321005.4956038,
dx : 6.9492501426787e-310,
abar : 17.391340262069377,
zbar : 8.012388199703627,
t_old : 6.9492501495557e-310,
dcvdt : 6.9492501259437e-310,
dcpdt : 6.94925012823816e-310,
ydot : array([ 3.79586303e-17, -2.15615792e-17, -1.37362275e-16,  1.25751835e-12,
    1.25920951e-13, -1.38340536e-12,  5.50015407e-19,  6.16893737e-38,
    1.34778166e-17,  3.79586303e-17,  4.98304614e-25, -1.20200533e+00,
   -1.61908807e+07]),
jac : array([[6.94925011e-310, 6.94925006e-310, 6.94925004e-310,
    6.94925002e-310, 6.94924997e-310, 6.94924995e-310,
    6.94925027e-310, 6.94925025e-310, 6.94925022e-310,
    6.94925018e-310, 6.94925015e-310, 6.94925013e-310,
    6.94925011e-310],
   [6.94925011e-310, 6.94925006e-310, 6.94925004e-310,
    6.94925002e-310, 6.94924997e-310, 6.94924995e-310,
    6.94925027e-310, 6.94925025e-310, 6.94925023e-310,
    6.94925018e-310, 6.94925016e-310, 6.94925013e-310,
    6.94925011e-310],
   [6.94925009e-310, 6.94925007e-310, 6.94925004e-310,
    6.94925000e-310, 6.94924997e-310, 6.94924995e-310,
    6.94925028e-310, 6.94925025e-310, 6.94925020e-310,
    6.94925018e-310, 6.94925016e-310, 6.94925014e-310,
    6.94925009e-310],
   [6.94925009e-310, 6.94925007e-310, 6.94925005e-310,
    6.94925000e-310, 6.94924998e-310, 6.94924995e-310,
    6.94925028e-310, 6.94930713e-310, 6.94925021e-310,
    6.94925018e-310, 6.94925016e-310, 6.94925011e-310,
    6.94925009e-310],
   [6.94925009e-310, 6.94925007e-310, 6.94925002e-310,
    6.94925000e-310, 6.94924998e-310, 6.94924993e-310,
    6.94925025e-310, 6.94925023e-310, 6.94925021e-310,
    6.94925019e-310, 6.94925014e-310, 6.94925011e-310,
    6.94925009e-310],
   [6.94925009e-310, 6.94925007e-310, 6.94925002e-310,
    6.94925000e-310, 6.94924998e-310, 0.00000000e+000,
    6.94925026e-310, 6.94925023e-310, 6.94925021e-310,
    6.94925019e-310, 6.94925014e-310, 6.94925012e-310,
    6.94925009e-310],
   [6.94925010e-310, 6.94925005e-310, 6.94925003e-310,
    6.94925000e-310, 6.94924996e-310, 1.97990385e-318,
    6.94925026e-310, 6.94925023e-310, 6.94925021e-310,
    6.94925016e-310, 6.94930713e-310, 6.94925012e-310,
    6.94925010e-310],
   [6.94925007e-310, 6.94925005e-310, 6.94925003e-310,
    6.94925001e-310, 6.94924996e-310, 4.65993225e-310,
```

```
        6.94925026e-310, 6.94925024e-310, 6.94925019e-310,
        6.94925017e-310, 6.94925014e-310, 6.94925012e-310,
        6.94925007e-310],
       [6.94925008e-310, 6.94925005e-310, 6.94925003e-310,
        6.94924998e-310, 6.94924996e-310, 6.94930722e-310,
        6.94925026e-310, 6.94925024e-310, 6.94925019e-310,
        6.94925017e-310, 6.94925014e-310, 6.94925012e-310,
        6.94925008e-310],
       [6.94925008e-310, 6.94925006e-310, 6.94925003e-310,
        6.94924999e-310, 6.94924996e-310, 0.00000000e+000,
        6.94925027e-310, 6.94925022e-310, 6.94925019e-310,
        6.94925017e-310, 6.94925015e-310, 6.94925010e-310,
        6.94925008e-310],
       [6.94925008e-310, 6.94925006e-310, 6.94925001e-310,
        6.94924999e-310, 6.94924997e-310, 0.00000000e+000,
        6.94925024e-310, 6.94925022e-310, 6.94925020e-310,
        6.94925017e-310, 6.94925015e-310, 6.94925010e-310,
        6.94925008e-310],
       [6.94925008e-310, 6.94925003e-310, 6.94925001e-310,
        6.94924999e-310, 6.94924997e-310, 6.94930712e-310,
        6.94925024e-310, 6.94925022e-310, 6.94925020e-310,
        6.94925015e-310, 6.94925013e-310, 6.94925010e-310,
        6.94925008e-310],
       [6.94925008e-310, 6.94925004e-310, 6.94925001e-310,
        6.94924999e-310, 6.94924994e-310, 7.31217156e-322,
        6.94925025e-310, 6.94925022e-310, 6.94925020e-310,
        6.94925015e-310, 6.94925013e-310, 6.94925011e-310,
        6.94925008e-310]]),
    self_heat : -1496870240,
    i : 32748,
    j : -1496823864,
    k : 32748,
    n_rhs : -1496777488,
    n_jac : 32748,
    time : 6.949250067493e-310,
    success : -1496684672}
```

```
In [25]: print("StarKiller-Maestro model core:")
         print("radius = {}".format(maestro_data.data('radius')[0]))
         print(sk_maestro_burn_results[0].state)
         sk_maestro_core = sk_maestro_burn_results[0]
```

```
StarKiller-Maestro model core:
radius = 9765.625
<burn_t>{
    rho : 8348566865.007088,
    t : 656585951.5581328,
```

```
e : 2.436851939389143e+18,
xn : array([1.00000046e-30, 8.92493058e-23, 5.99994547e-01, 3.92704949e-01,
   2.02111807e-09, 7.29698207e-03, 4.96502540e-16, 1.00000046e-30,
   3.52017152e-06, 1.00000046e-30, 1.00000046e-30]),
cv : 13444110.154561128,
cp : 13470469.479940196,
y_e : 0.4607295050374543,
eta : 133.98706481708632,
cs : 1086164797.2882276,
dx : 0.0,
abar : 17.391340173721446,
zbar : 8.012703550176676,
t_old : 0.0,
dcvdt : 0.0,
dcpdt : 8.4879959384e-314,
ydot : array([ 3.77137454e-17, -2.14245793e-17, -1.36464713e-16,  1.26306948e-12,
    1.21821687e-13, -1.38485747e-12,  5.60001897e-19,  6.29108323e-38,
    1.33908813e-17,  3.77137454e-17,  4.93356513e-25, -1.20443864e+00,
   -1.62243539e+07]),
jac : array([[6.42285340e-323, 1.26480805e-319, 0.00000000e+000,
    0.00000000e+000, 2.97079538e-313, 0.00000000e+000,
    0.00000000e+000, 1.26480805e-321, 0.00000000e+000,
    0.00000000e+000, 0.00000000e+000, 0.00000000e+000,
    6.94930722e-310],
   [1.65689855e-319, 0.00000000e+000, 0.00000000e+000,
    2.12200844e-314, 0.00000000e+000, 0.00000000e+000,
    4.24399286e-313, 0.00000000e+000, 0.00000000e+000,
    0.00000000e+000, 0.00000000e+000, 0.00000000e+000,
    6.94930722e-310],
   [0.00000000e+000, 0.00000000e+000, 3.18299503e-313,
    0.00000000e+000, 0.00000000e+000, 3.18299503e-313,
    0.00000000e+000, 0.00000000e+000, 2.12201236e-314,
    0.00000000e+000, 0.00000000e+000, 0.00000000e+000,
    6.94925013e-310],
   [0.00000000e+000, 2.12201615e-314, 0.00000000e+000,
    0.00000000e+000, 3.18299503e-313, 0.00000000e+000,
    0.00000000e+000, 4.45619244e-313, 0.00000000e+000,
    0.00000000e+000, 0.00000000e+000, 0.00000000e+000,
    6.94925013e-310],
   [2.97079525e-313, 0.00000000e+000, 0.00000000e+000,
    2.12201615e-314, 0.00000000e+000, 0.00000000e+000,
    2.97079538e-313, 0.00000000e+000, 0.00000000e+000,
    0.00000000e+000, 0.00000000e+000, 0.00000000e+000,
    6.94925013e-310],
   [0.00000000e+000, 0.00000000e+000, 3.39519447e-313,
    0.00000000e+000, 0.00000000e+000, 3.81959363e-313,
    0.00000000e+000, 9.88131292e-323, 2.12201615e-314,
    0.00000000e+000, 0.00000000e+000, 0.00000000e+000,
```

```
        6.94925013e-310],
       [0.00000000e+000, 6.36600015e-314, 0.00000000e+000,
        0.00000000e+000, 3.60739405e-313, 0.00000000e+000,
        0.00000000e+000, 1.69759866e-313, 0.00000000e+000,
        0.00000000e+000, 0.00000000e+000, 0.00000000e+000,
        6.94925013e-310],
       [2.12200857e-314, 0.00000000e+000, 0.00000000e+000,
        6.36600015e-314, 0.00000000e+000, 0.00000000e+000,
        2.12201236e-314, 0.00000000e+000, 0.00000000e+000,
        0.00000000e+000, 0.00000000e+000, 0.00000000e+000,
        6.37344683e-322],
       [0.00000000e+000, 0.00000000e+000, 2.12200857e-314,
        0.00000000e+000, 0.00000000e+000, 4.03179328e-313,
        0.00000000e+000, 0.00000000e+000, 1.26480805e-321,
        0.00000000e+000, 0.00000000e+000, 0.00000000e+000,
        6.94930722e-310],
       [6.91691904e-323, 3.16202013e-320, 0.00000000e+000,
        0.00000000e+000, 8.48799594e-314, 0.00000000e+000,
        0.00000000e+000, 4.66839202e-313, 0.00000000e+000,
        0.00000000e+000, 0.00000000e+000, 0.00000000e+000,
        6.94930722e-310],
       [1.90979824e-313, 0.00000000e+000, 7.41098469e-323,
        3.16202013e-320, 0.00000000e+000, 9.38724727e-323,
        2.12201615e-314, 0.00000000e+000, 0.00000000e+000,
        0.00000000e+000, 0.00000000e+000, 0.00000000e+000,
        6.94925014e-310],
       [0.00000000e+000, 0.00000000e+000, 1.90979824e-313,
        0.00000000e+000, 7.90505033e-323, 1.69759866e-313,
        0.00000000e+000, 0.00000000e+000, 2.54639638e-313,
        0.00000000e+000, 0.00000000e+000, 6.94925013e-310,
        6.94925014e-310],
       [0.00000000e+000, 2.97079538e-313, 0.00000000e+000,
        0.00000000e+000, 2.97079538e-313, 0.00000000e+000,
        0.00000000e+000, 2.97079538e-313, 6.94929490e-310,
        0.00000000e+000, 0.00000000e+000, 9.53546696e-322,
        6.94925014e-310]]),
  self_heat : -1495292992,
  i : 32748,
  j : -1495292632,
  k : 32748,
  n_rhs : -1495292272,
  n_jac : 32748,
  time : 6.94925013859575e-310,
  success : -1495291552}
```

In [26]:
```
def pdiff(x, xref):
    return 100.0*(x-xref)/xref
```

```
In [27]: print('density % diff = {}'.format(pdiff(sk_maestro_core.state.rho, sk_mesa_core.state.
         print('temperature % diff = {}'.format(pdiff(sk_maestro_core.state.t, sk_mesa_core.stat
         for i, (xmaestro, xmesa) in enumerate(zip(sk_maestro_core.state.xn, sk_mesa_core.state.
             print('xn({}) % diff = {}'.format(ecsn.short_species_names[i], pdiff(xmaestro, xmes
```

```
density % diff = -0.10197067618601745
temperature % diff = 0.0013822400974913848
xn(h1) % diff = inf
xn(he4) % diff = -0.4121021196308629
xn(o16) % diff = 2.3468657691855254e-06
xn(o20) % diff = -0.046158281066432334
xn(f20) % diff = 0.9181865326014039
xn(ne20) % diff = 2.5485216657050387
xn(mg24) % diff = 2.714129345751336
xn(al27) % diff = inf
xn(si28) % diff = -0.2303095616849742
xn(p31) % diff = inf
xn(s32) % diff = inf
```

```
/home/eugene/.local/lib/python3.7/site-packages/ipykernel_launcher.py:2: RuntimeWarning: divide
```