Decline Curve Models

This package provides model objects for decline curve models, and a means to attach a secondary GOR model to the primary phase model. As a requirement for inclusion in this package, all models are referenced to published petroleum engineering literature.

Use Examples

```
from pathlib import Path
import dca
from data import (rate, time)
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl

img_path = Path('../img')
plt.style.use('seaborn-white')
plt.rcParams['font.size'] = 16
```

Setup time series for Forecasts and calculate cumlative production of data

```
ftime = np.power(10, np.linspace(0, 4, 101))
data_N = np.cumsum(rate * [time[0], *np.diff(time)])
```

Time-Rate Decline Curve Models

Modified Hyperbolic Model

> Robertson, S. 1988. Generalized Hyperbolic Equation. Available from SPE, Richardson, Texas, USA. SPE-18731-MS.

```
mh = dca.MH(725, .85, .6, .2)
q_mh = mh.rate(ftime)
N_mh = mh.cum(ftime)
D_mh = mh.D(ftime)
b_mh = mh.b(ftime)
beta_mh = mh.beta(ftime)
N_mh *= data_N[-1] / mh.cum(time[-1])
```

Transient Hyperbolic Model

> Fulford, D. S., and Blasingame, T. A. 2013. Evaluation of Time-Rate Performance of Shale Wells using the Transient Hyperbolic Relation. Presented at SPE Unconventional Resources Conference – Canada in Calgary, Alberta, Canda, 5–7 November. SPE-167242-MS. https://doi.org/10.2118/167242-MS.

```
thm = dca.THM(750, .8, 2, .5, 28)
q_thm = thm.rate(ftime)
N_thm = thm.cum(ftime)
D_thm = thm.D(ftime)
b_thm = thm.b(ftime)
beta_thm = thm.beta(ftime)
N_thm *= data_N[-1] / thm.cum(time[-1])
```

Power-Law Exponential Model

> Ilk, D., Perego, A. D., Rushing, J. A., and Blasingame, T. A. 2008. Exponential vs. Hyperbolic Decline in Tight Gas Sands – Understanding the Origin and Implications for Reserve Estimates Using Arps Decline Curves. Presented at SPE Annual Technical Conference and Exhibition in Denver, Colorado, USA, 21–24 September. SPE-116731-MS. https://doi.org/10.2118/116731-MS.

> Ilk, D., Rushing, J. A., and Blasingame, T. A. 2009. Decline Curve Analysis for HP/HT Gas Wells: Theory and Applications. Presented at SPE Annual Technical Conference and Exhibition in New Orleands, Louisiana, USA, 4–7 October. SPE-125031-MS. https://doi.org/10.2118/125031-MS.

```
ple = dca.PLE(750, .1, .00001, .5)
q_ple = ple.rate(ftime)
N_ple = ple.cum(ftime)
D_ple = ple.D(ftime)
b_ple = ple.b(ftime)
beta_ple = ple.beta(ftime)
N_ple *= data_N[-1] /  ple.cum(time[-1])
```

Stretched Exponential

> Valkó, P. P. Assigning Value to Stimulation in the Barnett Shale: A Simultaneous Analysis of 7000 Plus Production Histories and Well Completion Records. 2009. Presented at SPE Hydraulic Fracturing Technology Conference in College Station, Texas, USA, 19–21 January. SPE-119369-MS. https://doi.org/10.2118/119369-MS.

```
se = dca.SE(715, 90.0, .5)
q_se = se.rate(ftime)
N_se = se.cum(ftime)
D_se = se.D(ftime)
b_se = se.b(ftime)
beta_se = se.beta(ftime)
N_se *= data_N[-1] / se.cum(time[-1])
```

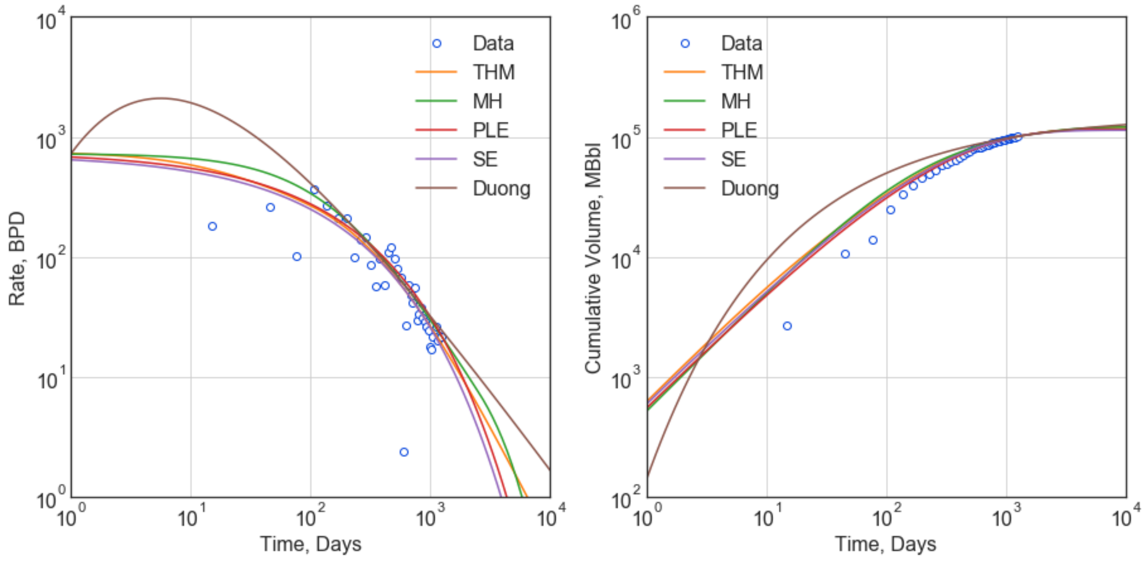
Duong Model

> Duong, A. N. 2001. Rate-Decline Analysis for Fracture-Dominated Shale Reservoirs. SPE Res Eval & Eng 14 (3): 377–387. SPE-137748-PA. https://doi.org/10.2118/137748-PA.

```
dg = dca.Duong(715, 2.8, 1.4)
q_dg = dg.rate(ftime)
N_dg = dg.cum(ftime)
D_dg = dg.D(ftime)
b_dg = dg.b(ftime)
beta_dg = dg.beta(ftime)
N_dg *= data_N[-1] / dg.cum(time[-1])
```

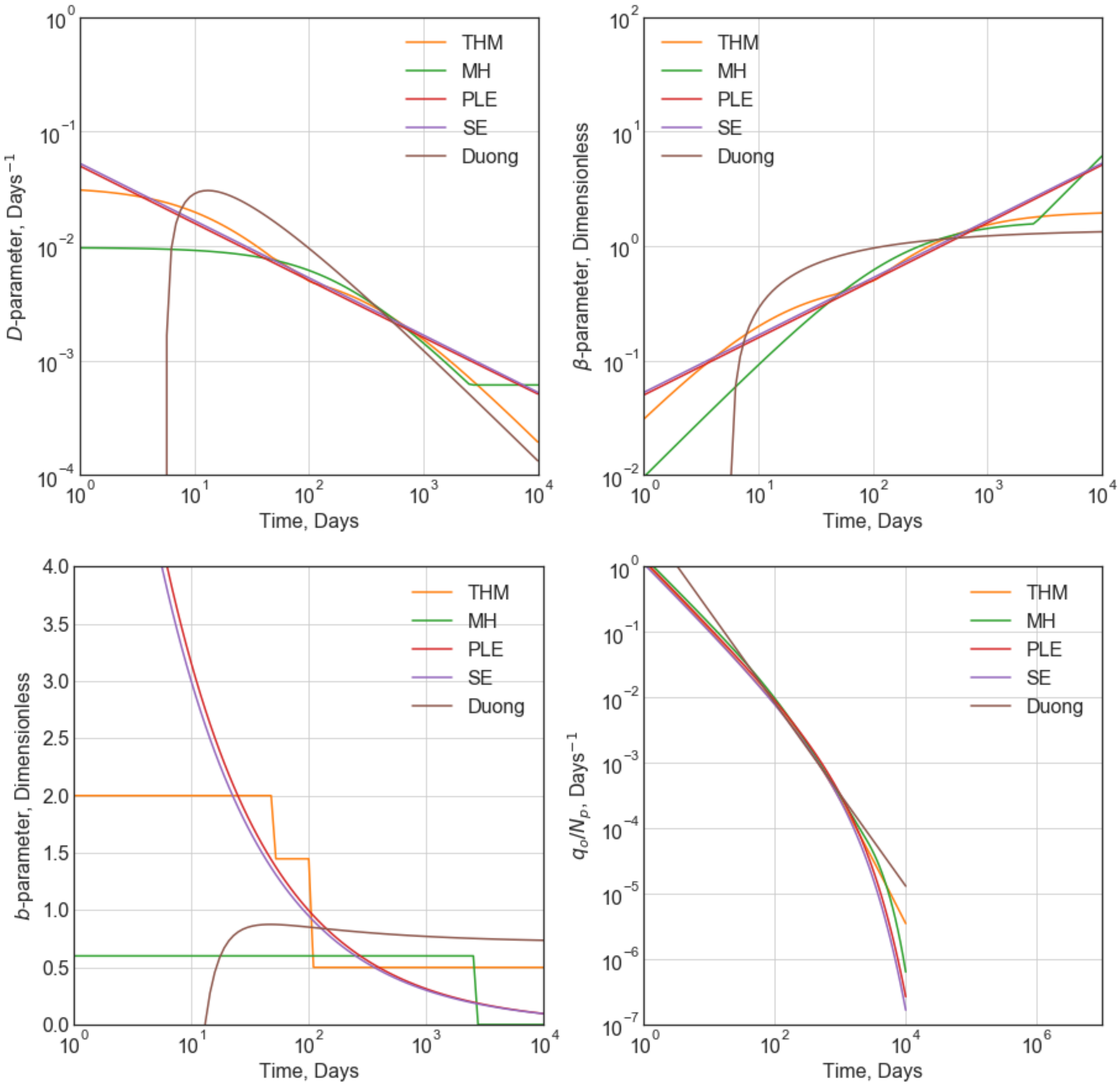
Time-Rate Model Diagnostic Plots

```
# Rate vs Time
fig = plt.figure(figsize=(15, 7.5))
ax1 = fig.add_subplot(121)
ax2 = fig.add_subplot(122)

ax1.loglog(time, rate, 'o', mfc='w', label='Data')
ax1.loglog(ftime, q_thm, label='THM')
ax1.loglog(ftime, q_mh, label='MH')
ax1.loglog(ftime, q_ple, label='PLE')
ax1.loglog(ftime, q_se, label='SE')
ax1.loglog(ftime, q_dg, label='Duong')

ax1.set(ylabel='Rate, BPD', xlabel='Time, Days')
ax1.set(ylim=(1e0, 1e4), xlim=(1e0, 1e4))
ax1.set_aspect(1)
ax1.grid()
ax1.legend()

ax2.loglog(time, data_N, 'o', mfc='w', label='Data')
ax2.loglog(ftime, N_thm, label='THM')
ax2.loglog(ftime, N_mh, label='MH')
ax2.loglog(ftime, N_ple, label='PLE')
ax2.loglog(ftime, N_se, label='SE')
ax2.loglog(ftime, N_dg, label='Duong')

ax2.set(ylim=(1e2, 1e6), xlim=(1e0, 1e4))
ax2.set(ylabel='Cumulative Volume, MBbl', xlabel='Time, Days')
ax2.set_aspect(1)
ax2.grid()
ax2.legend()

plt.savefig(img_path / 'model.png')
```




```
fig = plt.figure(figsize=(15, 15))
ax1 = fig.add_subplot(221)
ax2 = fig.add_subplot(222)
ax3 = fig.add_subplot(223)
ax4 = fig.add_subplot(224)

ax1.loglog([], [])
ax1.loglog(ftime, D_thm, label='THM')
ax1.loglog(ftime, D_mh, label='MH')
ax1.loglog(ftime, D_ple, label='PLE')
ax1.loglog(ftime, D_se, label='SE')
ax1.loglog(ftime, D_dg, label='Duong')
ax1.set(ylim=(1e-4, 1e0))
ax1.set(ylabel='$D$-parameter, Days$^{-1}$', xlabel='Time, Days')

ax2.loglog([], [])
ax2.loglog(ftime, beta_thm, label='THM')
ax2.loglog(ftime, beta_mh, label='MH')
ax2.loglog(ftime, beta_ple, label='PLE')
ax2.loglog(ftime, beta_se, label='SE')
ax2.loglog(ftime, beta_dg, label='Duong')
ax2.set(ylim=(1e-2, 1e2))
ax2.set(ylabel=r'$\beta$-parameter, Dimensionless', xlabel='Time, Days')

ax3.semilogx([], [])
ax3.semilogx(ftime, b_thm, label='THM')
ax3.semilogx(ftime, b_mh, label='MH')
ax3.semilogx(ftime, b_ple, label='PLE')
ax3.semilogx(ftime, b_se, label='SE')
ax3.semilogx(ftime, b_dg, label='Duong')
ax3.set(ylim=(0., 4.))
ax3.set(ylabel='$b$-parameter, Dimensionless', xlabel='Time, Days')
```

```
ax4.loglog([], [])
ax4.loglog(ftime, q_thm / N_thm, label='THM')
ax4.loglog(ftime, q_mh / N_mh, label='MH')
ax4.loglog(ftime, q_ple / N_ple, label='PLE')
ax4.loglog(ftime, q_se / N_se, label='SE')
ax4.loglog(ftime, q_dg / N_dg, label='Duong')
ax4.set(ylim=(1e-7, 1e0), xlim=(1e0, 1e7))
ax4.set(ylabel='$q_o / N_p$, Days$^{-1}$', xlabel='Time, Days')

for ax in [ax1, ax2, ax3, ax4]:
    if ax != ax4:
        ax.set(xlim=(1e0, 1e4))
    if ax != ax3:
        ax.set_aspect(1)
    ax.grid()
    ax.legend()


plt.savefig(img_path / 'diagnostics.png')
```



GOR/CGR Model

Power-Law GOR/CGR Model.

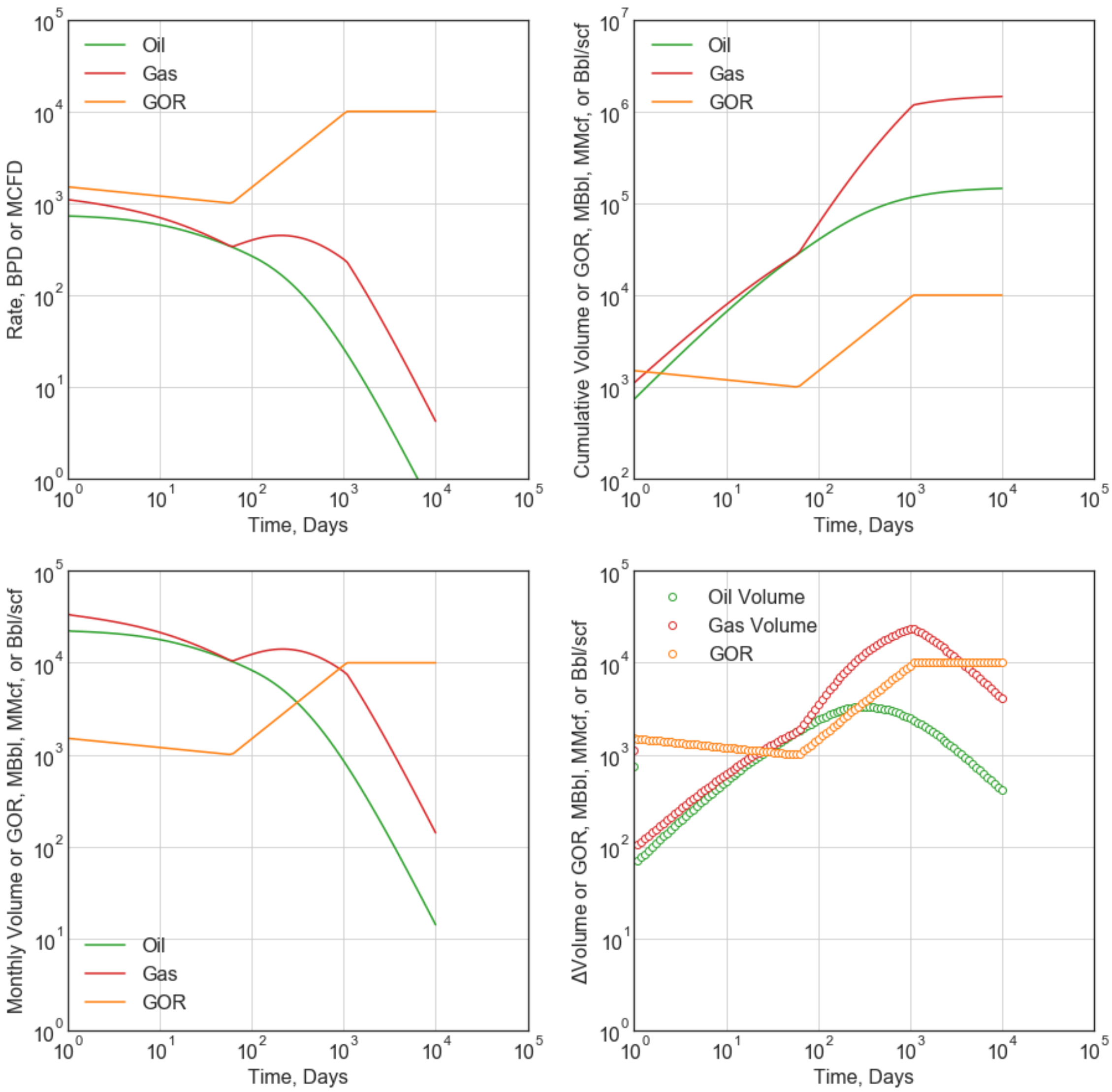
> Fulford, D.S. 2018. A Model-Based Diagnostic Workflow for Time-Rate Performance of Unconventional Wells. Presented at Unconventional Resources Conference in Houston, Texas, USA, 23–25 July. URTeC-2903036. https://doi.org/10.15530/urtec-2018-2903036.

```
thm.add_secondary(c=1000, m0=-.1, m=.8, t0=2 * 365.25 / 12, max=10_000)
```

GOR/CGR Time-Rate Diagnostic Plots

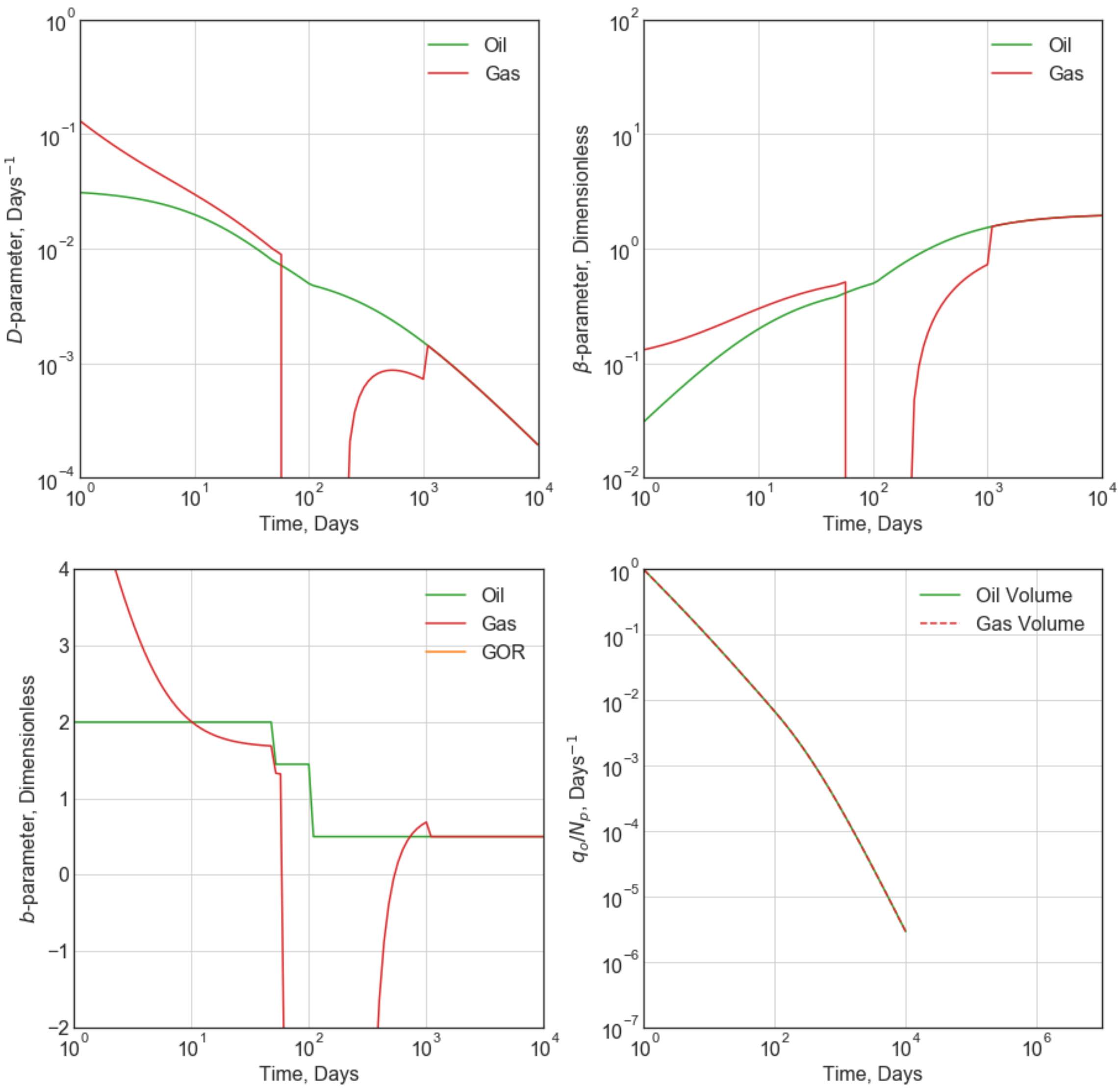
```
fig = plt.figure(figsize=(15, 15))
ax1 = fig.add_subplot(221)
ax2 = fig.add_subplot(222)
ax3 = fig.add_subplot(223)
ax4 = fig.add_subplot(224)


# Rate vs Time

q = thm.rate(ftime)
g = thm.secondary.rate(ftime)
y = thm.secondary.gor(ftime)

ax1.plot(ftime, q, c='C2', label='Oil')
ax1.plot(ftime, g, c='C3', label='Gas')
ax1.plot(ftime, y, c='C1', label='GOR')
ax1.set(xscale='log', yscale='log', xlim=(1e0, 1e5), ylim=(1e0, 1e5))
ax1.set(ylabel='Rate, BPD or MCFD', xlabel='Time, Days')


# Cumulative Volume vs Time

q = thm.cum(ftime)
g = thm.secondary.cum(ftime)
y = thm.secondary.gor(ftime)

ax2.plot(ftime, q, c='C2', label='Oil')
ax2.plot(ftime, g, c='C3', label='Gas')
ax2.plot(ftime, y, c='C1', label='GOR')
ax2.set(xscale='log', yscale='log', xlim=(1e0, 1e5), ylim=(1e2, 1e7))
ax2.set(ylabel='Rate, Dimensionless', xlabel='Time, Days')
ax2.set(ylabel='Cumulative Volume or GOR, MBbl, MMcf, or Bbl/scf', xlabel='Time, Days')


# Time vs Monthly Volume

q = thm.monthly_vol(ftime)
g = thm.secondary.monthly_vol(ftime)
y = thm.secondary.gor(ftime)

ax3.plot(ftime, q, c='C2', label='Oil')
ax3.plot(ftime, g, c='C3', label='Gas')
ax3.plot(ftime, y, c='C1', label='GOR')
ax3.set(xscale='log', yscale='log', xlim=(1e0, 1e5), ylim=(1e0, 1e5))
ax3.set(ylabel='Monthly Volume or GOR, MBbl, MMcf, or Bbl/scf', xlabel='Time, Days')


# Time vs Interval Volume

q = thm.interval_vol(ftime, t0=0.)
g = thm.secondary.interval_vol(ftime, t0=0.)
```

```
y = thm.secondary.gor(ftime)

ax4.plot(ftime, q, 'o', mec='C2', mfc='white', label='Oil Volume')
ax4.plot(ftime, g, 'o', mec='C3', mfc='white', label='Gas Volume')
ax4.plot(ftime, y, 'o', mec='C1', mfc='white', label='GOR')
ax4.set(xscale='log', yscale='log', xlim=(1e0, 1e5), ylim=(1e0, 1e5))
ax4.set(ylabel='$\Delta$Volume or GOR, MBbl, MMcf, or Bbl/scf', xlabel='Time,
Days')

for ax in [ax1, ax2, ax3, ax4]:
    ax.set_aspect(1)
    ax.grid()
    ax.legend()

plt.savefig(img_path / 'secondary_model.png')
```



```
fig = plt.figure(figsize=(15, 15))
ax1 = fig.add_subplot(221)
ax2 = fig.add_subplot(222)
ax3 = fig.add_subplot(223)
ax4 = fig.add_subplot(224)


# Rate vs Time

q = thm.D(ftime)
g = thm.secondary.D(ftime)

ax1.plot(ftime, q, c='C2', label='Oil')
ax1.plot(ftime, g, c='C3', label='Gas')
ax1.set(xscale='log', yscale='log', xlim=(1e0, 1e4), ylim=(1e-4, 1e0))
ax1.set(ylabel='$D$-parameter, Days$^{-1}$', xlabel='Time, Days')


# Cumulative Volume vs Time

q = thm.beta(ftime)
g = thm.secondary.beta(ftime)

ax2.plot(ftime, q, c='C2', label='Oil')
ax2.plot(ftime, g, c='C3', label='Gas')
ax2.set(xscale='log', yscale='log', xlim=(1e0, 1e4), ylim=(1e-2, 1e2))
ax2.set(ylabel=r'$\beta$-parameter, Dimensionless', xlabel='Time, Days')


# Time vs Monthly Volume

q = thm.b(ftime)
g = thm.secondary.b(ftime)

ax3.plot(ftime, q, c='C2', label='Oil')
ax3.plot(ftime, g, c='C3', label='Gas')
ax3.plot(ftime, y, c='C1', label='GOR')
ax3.set(xscale='log', yscale='linear', xlim=(1e0, 1e4), ylim=(-2, 4))
ax3.set(ylabel='$b$-parameter, Dimensionless', xlabel='Time, Days')


# Time vs Interval Volume

q = thm.rate(ftime) / thm.cum(ftime)
g = thm.secondary.rate(ftime) / thm.secondary.cum(ftime)

ax4.plot(ftime, q, c='C2', label='Oil Volume')
ax4.plot(ftime, g, c='C3', ls='--', label='Gas Volume')
ax4.set(xscale='log', yscale='log', ylim=(1e-7, 1e0), xlim=(1e0, 1e7))
ax4.set(ylabel='$q_o / N_p$, Days$^{-1}$', xlabel='Time, Days')

for ax in [ax1, ax2, ax3, ax4]:
    if ax != ax3:
```

```
        ax.set_aspect(1)
    ax.grid()
    ax.legend()

plt.savefig(img_path / 'sec_diagnostic_funs.png')
```



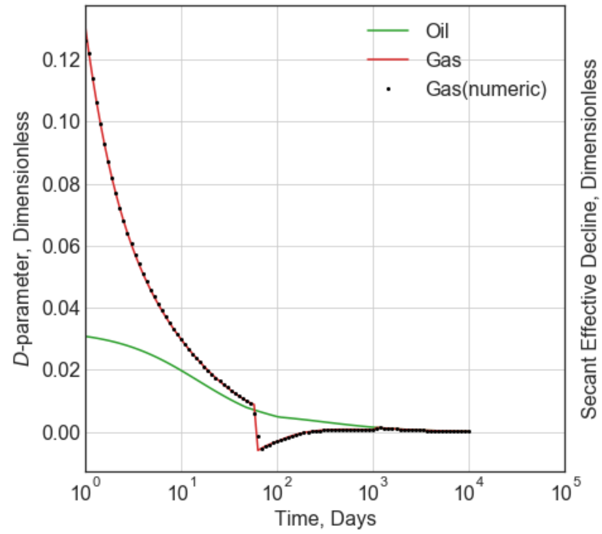
Additional Diagnostics

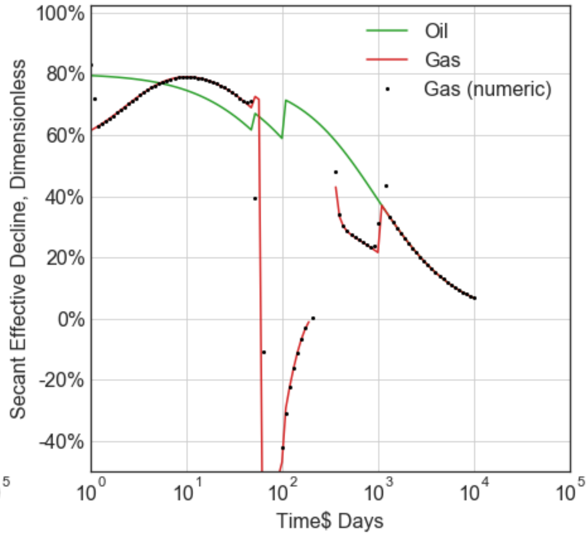
Numeric calculation provided to verify analytic relationships

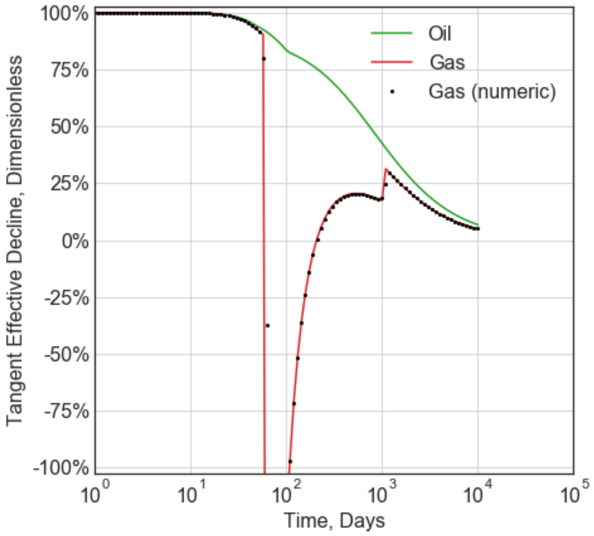
```
fig = plt.figure(figsize=(15, 15))
ax1 = fig.add_subplot(221)
ax2 = fig.add_subplot(222)
ax3 = fig.add_subplot(223)
```

```
# D-parameter vs Time

q = thm.D(ftime)
g = thm.secondary.D(ftime)
_g = -np.gradient(np.log(thm.secondary.rate(ftime)), ftime)

ax1.plot(ftime, q, c='C2', label='Oil')
ax1.plot(ftime, g, c='C3', label='Gas')
ax1.plot(ftime, _g, 'o', ms=2, c='k', label='Gas(numeric)')
ax1.set(xscale='log', yscale='linear', xlim=(1e0, 1e5), ylim=(None, None))
ax1.set(ylabel='$D$-parameter, Dimensionless', xlabel='Time, Days')


# Secant Effective Decline vs Time

secant_from_nominal = dca.MultisegmentHyperbolic.secant_from_nominal

qn = [secant_from_nominal(d * 365.25, b) for d, b in zip(q, thm.b(ftime))]
gn = [secant_from_nominal(d * 365.25, b) for d, b in zip(g,
thm.secondary.b(ftime))]
_gn = [secant_from_nominal(d * 365.25, b) for d, b in zip(g, np.gradient(1 / _g,
ftime))]

ax2.plot(ftime, qn, c='C2', label='Oil')
ax2.plot(ftime, gn, c='C3', label='Gas')
ax2.plot(ftime, _gn, 'o', c='k', ms=2, label='Gas (numeric)')
ax2.set(xscale='log', yscale='linear', xlim=(1e0, 1e5), ylim=(-.5, 1.025))
ax2.yaxis.set_major_formatter(mpl.ticker.PercentFormatter(xmax=1))
ax2.set(ylabel='Secant Effective Decline, Dimensionless', xlabel='Time$ Days')


# Tangent Effective Decline vs Time

ax3.plot(ftime, 1 - np.exp(-q * 365.25), c='C2', label='Oil')
ax3.plot(ftime, 1 - np.exp(-g * 265.25), c='C3', label='Gas')
ax3.plot(ftime, 1 - np.exp(-_g * 265.25), 'o', c='k', ms=2, label='Gas (numeric)')
ax3.set(xscale='log', yscale='linear', xlim=(1e0, 1e5), ylim=(-1.025, 1.025))
ax3.yaxis.set_major_formatter(mpl.ticker.PercentFormatter(xmax=1))
ax3.set(ylabel='Tangent Effective Decline, Dimensionless', xlabel='Time, Days')

for ax in [ax1, ax2, ax3]:
    ax.grid()
    ax.legend()

plt.savefig(img_path / 'sec_decline_diagnostics.png')
```

```
c:\users\dfulford\projects\dca\dca\dca.py:693: RuntimeWarning: invalid value
encountered in double_scalars
  return 1.0 - 1.0 / (1.0 + D * b) ** (1.0 / b)
```


Oil
Gas
Gas(numeric)
D-parameter, Dimensionless
0.12
0.10
0.08
0.06
0.04
0.02
0.00
10^0
10^1
10^2
10^3
10^4
10^5
Time, Days


Oil
Gas
Gas (numeric)
Secant Effective Decline, Dimensionless
100%
80%
60%
40%
20%
0%
-20%
-40%
10^0
10^1
10^2
10^3
10^4
10^5
Time$ Days


Oil
Gas
Gas (numeric)
Tangent Effective Decline, Dimensionless
100%
75%
50%
25%
0%
-25%
-50%
-75%
-100%
10^0
10^1
10^2
10^3
10^4
10^5
Time, Days