Jacobian_approach(mu_a)_old

January 17, 2024

1 Determine Jacobinas(DelI for Del TMP) Analytically and Plot Them

2 Values Ranges Used in Simulation (New)

Variable	Range(lower)	Range(Upper)	Point Count
Materna Sturation	0.9	1.0	5
Maternal Hb Conc	11	15	5
Fetal Saturation	0.2	0.6	5
Fetal Hb Conc	11	15	5
Maternal BVF	0.2		
Fetal BVF	0.22		

3 Values Ranges Used in Simulation (old)

Variable	Range(lower)	Range(Upper)	Point Count
Materna Sturation	0.9	1.0	5
Maternal Hb Conc	11	15	5
Fetal Saturation	0.1	0.6	5
Fetal Hb Conc	0.11	0.15	5

[ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from inverse_modelling_tfo.tools.s_based_intensity_datagen import MU_MAP_BASE1,␣
↪MU_MAP_BASE2
from tfo_sensitivity.jacobian import (
    MuANumericalJC,
    FullBloodAnalyticalJC,
    FullBloodJacobianMuAEqn,
    PartialBloodAnalyticalJC,
    PartialBloodJacobianMuAEqn,
```

```
    OperatingPoint,
)

# Plotting
FIG_WIDTH = 8
FIG_HEIGHT = 4
plt.style.use("seaborn")
# plt.rcParams['figure.dpi'] = 150    # Smaller plot
plt.rcParams["figure.dpi"] = 700  # Paper-ready plots

# Loading Files
maternal_wall_thickness, uterus_thickness, wave_int = 20, 5, 1
base_mu_map = MU_MAP_BASE1 if wave_int == 1 else MU_MAP_BASE2
```

4 Defining Base Parameters

[ ]:
```
# Base Parameters
MATERNAL_Hb = 11.
MATERNAL_SAT = 1.0
FETAL_SAT = 0.50
FETAL_Hb = 0.11
# FETAL_Hb = 11.0

# Sweep Parameters
# all_fetal_c = np.linspace(11, 16, 6)
all_fetal_c = np.linspace(0.11, 0.16, 6)

all_fetal_sat = np.linspace(0.1, 0.6, 6)

PLOT_NORMALIZED = True  # Plot the Jacibian divided by Current Intensity␣
↪(Similar to normalized derivative)

# Jacobian Calculator
mu_a_eqn = FullBloodJacobianMuAEqn()    # How the mu_a is calculated for Fetal/
↪Maternal variable layers
# mu_a_eqn = PartialBloodJacobianMuAEqn(0.2, 0.1, 0.75, 0.2, 0.1, 0.75)    #␣
↪How the mu_a is calculated for Fetal/Maternal variable layers
```

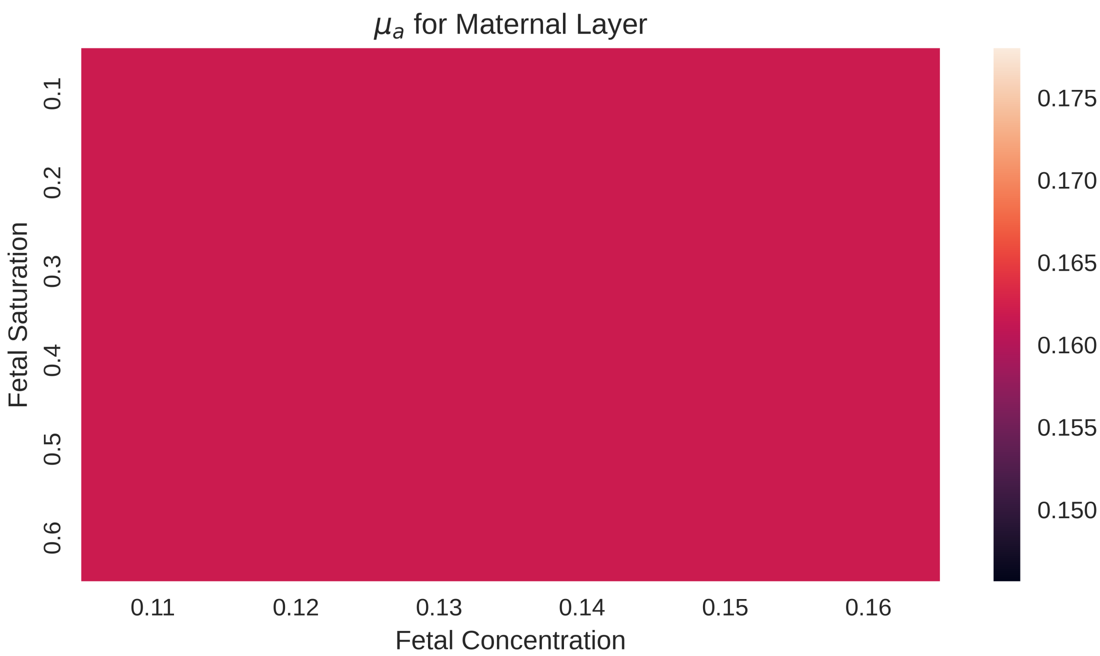
5 Calculating Values of μ_a

[ ]:
```
mu_data_table = pd.DataFrame(columns=["Fetal Saturation", "Fetal␣
↪Concentration", "MuA", "Type"])
# Round 1 - Plots for Varying Fetal Saturation (type 1 & 3)
for fs in all_fetal_sat:
```

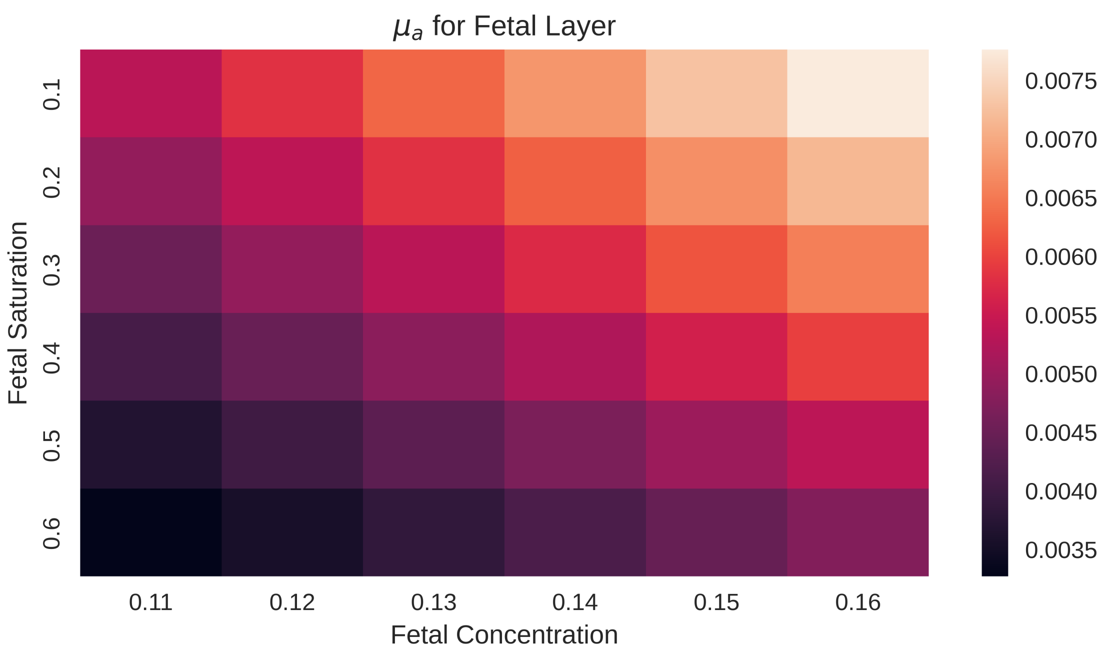
```python
    fs = round(fs, 2)  # np.range sometimes creates weird numbers... round to 2␣
↪decimal places
    # Round 2 - Plots for Varying Fetal Concentration (type 2 & 4)
    for fc in all_fetal_c:
        fc = round(fc, 2)  # np.range sometimes creates weird numbers... round␣
↪to 2 decimal places
        operating_point = OperatingPoint(MATERNAL_Hb, MATERNAL_SAT, fc, fs,␣
↪wave_int)

        mu_a = mu_a_eqn.get_mu_map(base_mu_map, operating_point)

        # Adding to Table
        new_row1 = {
            "Fetal Saturation": fs,
            "Fetal Concentration": fc,
            "MuA": mu_a[1],
            "Type": "mom"
        }
        new_row2 = {
            "Fetal Saturation": fs,
            "Fetal Concentration": fc,
            "MuA": mu_a[4],
            "Type": "fetus"
        }
        mu_data_table.loc[len(mu_data_table)] = new_row1
        mu_data_table.loc[len(mu_data_table)] = new_row2
```

[ ]:
```python
# Plot Mu (Mom & Fetus)
mom_data_table : pd.DataFrame = mu_data_table[mu_data_table['Type'] == 'mom']
fetus_data_table : pd.DataFrame= mu_data_table[mu_data_table['Type'] == 'fetus']

plt.figure(figsize=(FIG_WIDTH, FIG_HEIGHT))
sns.heatmap(mom_data_table.pivot(index='Fetal Saturation', columns='Fetal␣
↪Concentration', values='MuA'))
plt.title('$\mu_a$ for Maternal Layer')

plt.figure(figsize=(FIG_WIDTH, FIG_HEIGHT))
sns.heatmap(fetus_data_table.pivot(index='Fetal Saturation', columns='Fetal␣
↪Concentration', values='MuA'))
plt.title('$\mu_a$ for Fetal Layer')
```

[ ]:
```
Text(0.5, 1.0, '$\\mu_a$ for Fetal Layer')
```



μ_a for Maternal Layer
0.1
0.2
0.3
0.4
0.5
0.6
Fetal Saturation
0.11
0.12
0.13
0.14
0.15
0.16
Fetal Concentration
0.175
0.170
0.165
0.160
0.155
0.150



μ_a for Fetal Layer
0.1
0.2
0.3
0.4
0.5
0.6
Fetal Saturation
0.11
0.12
0.13
0.14
0.15
0.16
Fetal Concentration
0.0075
0.0070
0.0065
0.0060
0.0055
0.0050
0.0045
0.0040
0.0035

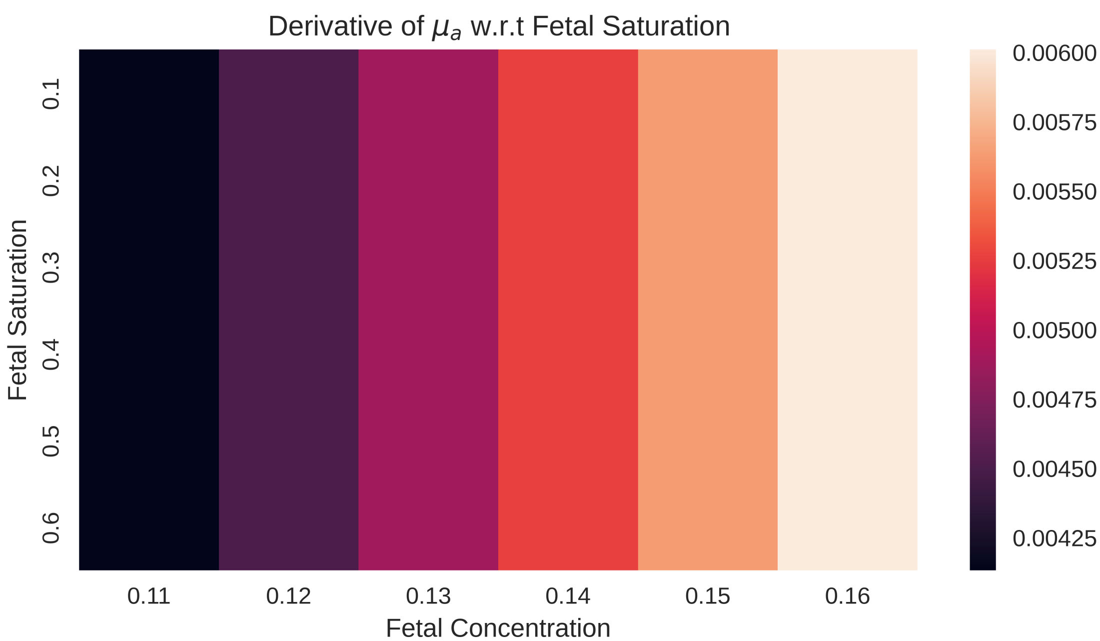
6 Calculating Derivatives

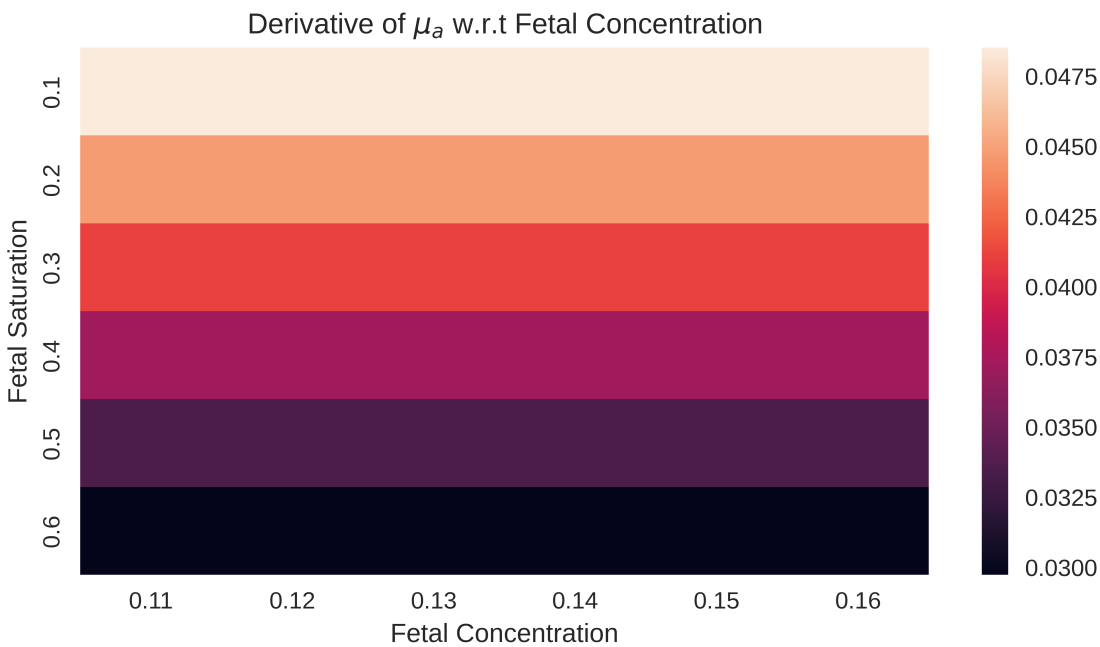
```
[ ]: data_table = pd.DataFrame(columns=["Fetal Saturation", "Fetal Concentration",␣
     ↪"Derivative", "Type"])
      # Types = 1, 2, 3, 4

     # Comment on Types: 1 & 2 : delI/delFS, 3 & 4 : delI/delFC
     # Types 1 & 3: Fetal Sat varies, Types 2 & 4: Fetal Conc varies


     # Round 1 - Plots for Varying Fetal Saturation (type 1 & 3)
     for fs in all_fetal_sat:
         fs = round(fs, 2)  # np.range sometimes creates weird numbers... round to 2␣
     ↪decimal places
         # Round 2 - Plots for Varying Fetal Concentration (type 2 & 4)
         for fc in all_fetal_c:
             fc = round(fc, 2)  # np.range sometimes creates weird numbers... round␣
     ↪to 2 decimal places

             operating_point = OperatingPoint(MATERNAL_Hb, MATERNAL_SAT, fc, fs,␣
     ↪wave_int)

             AnalyticalJC = MuANumericalJC(operating_point, "FS", mu_a_eqn)
             numerical_term1 = AnalyticalJC.calculate_jacobian()

             AnalyticalJC = MuANumericalJC(operating_point, "FC", mu_a_eqn)
             numerical_term2 = AnalyticalJC.calculate_jacobian()

             # Adding to Table
             new_row1 = {
                 "Fetal Saturation": fs,
                 "Fetal Concentration": fc,
                 "Derivative": numerical_term1,
                 "Type": "FS",
             }
             new_row2 = {
                 "Fetal Saturation": fs,
                 "Fetal Concentration": fc,
                 "Derivative": numerical_term2,
                 "Type": "FC",
             }
             data_table.loc[len(data_table)] = new_row1
             data_table.loc[len(data_table)] = new_row2
```

7 Plotting Data

[ ]:
```python
# Mandatory - Make all the derivatives positive
data_table['Derivative'] = data_table['Derivative'].abs()

fs_data_table : pd.DataFrame = data_table[data_table['Type'] == 'FS']
fc_data_table : pd.DataFrame= data_table[data_table['Type'] == 'FC']

plt.figure(figsize=(FIG_WIDTH, FIG_HEIGHT))
sns.heatmap(fs_data_table.pivot(index='Fetal Saturation', columns='Fetal␣
 ↪Concentration', values='Derivative'))
plt.title('Derivative of $\mu_a$ w.r.t Fetal Saturation')

plt.figure(figsize=(FIG_WIDTH, FIG_HEIGHT))
sns.heatmap(fc_data_table.pivot(index='Fetal Saturation', columns='Fetal␣
 ↪Concentration', values='Derivative'))
plt.title('Derivative of $\mu_a$ w.r.t Fetal Concentration')
```

[ ]:
```
Text(0.5, 1.0, 'Derivative of $\\mu_a$ w.r.t Fetal Concentration')
```





Derivative of μ_a w.r.t Fetal Concentration
Fetal Saturation
0.1
0.2
0.3
0.4
0.5
0.6
0.11
0.12
0.13
0.14
0.15
0.16
Fetal Concentration
0.0475
0.0450
0.0425
0.0400
0.0375
0.0350
0.0325
0.0300