display_voltage

May 8, 2018

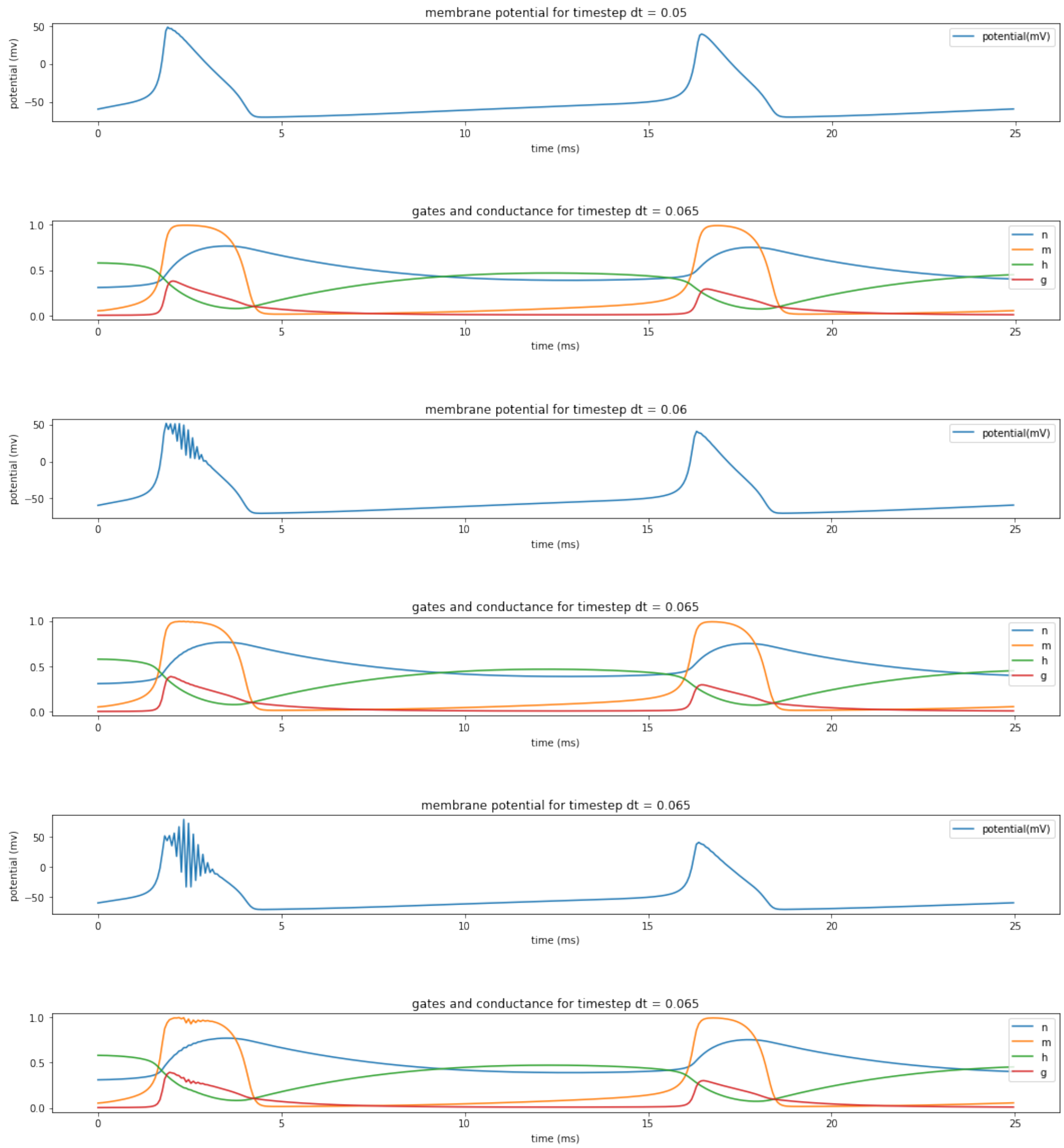
0.0.1 Conductance dynamics and effects on Forward Euler Method Stability

We validates by implementation the theoretical results on minimum time step found previously.

We also demonstrate that minimum time step depends on conductance value: the solution begins to be unstable when conductance reach a critical value.

```
In [17]: import seaborn as sns
         import matplotlib.pyplot as plt
         import matplotlib as matplotlib
         import pandas as pd
         matplotlib.rcParams['figure.figsize'] = [18, 20]
         df_006 = pd.read_csv("/home/fouriaux/Devel/reaching_ftb/hh/data/fe_dt_006.csv")
         potential_006 = df_006[["time(ms)", "potential(mV)"]].copy()
         potential_006 = potential_006.set_index("time(ms)")
         gates_006 = df_006[["time(ms)", "n","m","h", "g"]]
         gates_006 = gates_006.set_index("time(ms)")

         df_0065 = pd.read_csv("/home/fouriaux/Devel/reaching_ftb/hh/data/fe_dt_0065.csv")
         potential_0065 = df_0065[["time(ms)", "potential(mV)"]].copy()
         potential_0065 = potential_0065.set_index("time(ms)")
         gates_0065 = df_0065[["time(ms)", "n","m","h", "g"]]
         gates_0065 = gates_0065.set_index("time(ms)")

         df_005 = pd.read_csv("/home/fouriaux/Devel/reaching_ftb/hh/data/fe_dt_005.csv")
         potential_005 = df_005[["time(ms)", "potential(mV)"]].copy()
         potential_005 = potential_005.set_index("time(ms)")
         gates_005 = df_005[["time(ms)", "n","m","h", "g"]]
         gates_005 = gates_005.set_index("time(ms)")
         plt.subplots_adjust(hspace=1)
         plt.subplot(6,1,1)
         header = list(potential_005)
         hh_v = plt.plot (potential_005)
         plt.legend(iter(hh_v), header)
         plt.title("membrane potential for timestep dt = 0.05")
         plt.xlabel("time (ms)")
         plt.ylabel("potential (mv)")
         plt.subplot(6,1,2)
         header = list(gates_005)
```

```
plt.title("gates and conductance for timestep dt = 0.065")
hh_v = plt.plot (gates_005)
plt.legend(iter(hh_v), header)
plt.xlabel("time (ms)")


plt.subplot(6,1,3)
header = list(potential_006)
hh_v = plt.plot (potential_006)
plt.legend(iter(hh_v), header)
plt.title("membrane potential for timestep dt = 0.06")
plt.xlabel("time (ms)")
plt.ylabel("potential (mv)")
plt.subplot(6,1,4)
header = list(gates_006)
plt.title("gates and conductance for timestep dt = 0.065")
hh_v = plt.plot (gates_006)
plt.legend(iter(hh_v), header)
plt.xlabel("time (ms)")

plt.subplot(6,1,5)
header = list(potential_0065)
hh_v = plt.plot (potential_0065)
plt.legend(iter(hh_v), header)
plt.title("membrane potential for timestep dt = 0.065")
plt.xlabel("time (ms)")
plt.ylabel("potential (mv)")
plt.subplot(6,1,6)
header = list(gates_0065)
plt.title("gates and conductance for timestep dt = 0.065")
hh_v = plt.plot (gates_0065)
plt.legend(iter(hh_v), header)
plt.xlabel("time (ms)")


plt.show()
```

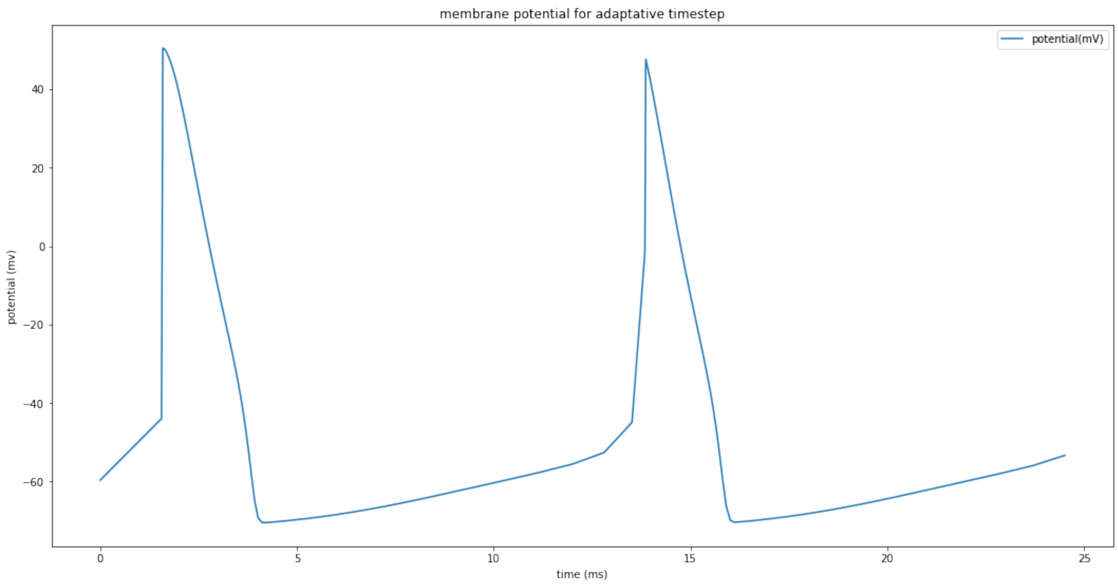


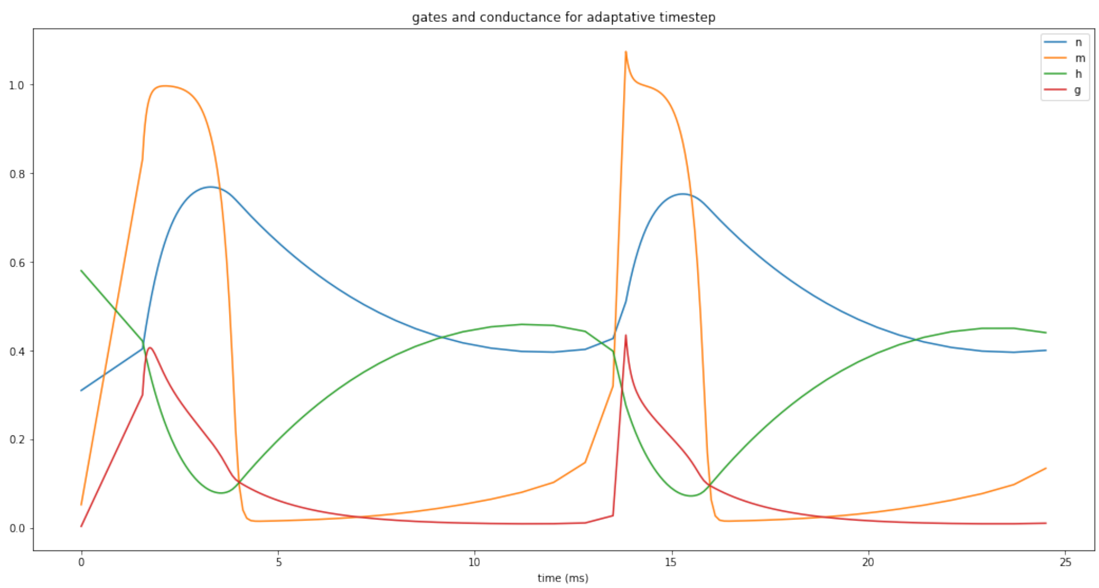
0.0.2 Adaptative Forward Euler time step

to be refined

```
In [23]: df = pd.read_csv("/home/fouriaux/Devel/reaching_ftb/hh/data/afe.csv")
         potential = df[["time(ms)", "potential(mV)"]].copy()
         potential = potential.set_index("time(ms)")
         gates = df[["time(ms)", "n","m","h", "g"]]
         gates = gates.set_index("time(ms)")
         plt.subplots_adjust(hspace=0.2)
         plt.subplot(2,1,1)
```

```
header = list(potential)
hh_v = plt.plot (potential)
plt.legend(iter(hh_v), header)
plt.title("membrane potential for adaptative timestep")
plt.xlabel("time (ms)")
plt.ylabel("potential (mv)")
plt.subplot(2,1,2)
header = list(gates)
plt.title("gates and conductance for adaptative timestep")
hh_v = plt.plot (gates)
plt.legend(iter(hh_v), header)
plt.xlabel("time (ms)")
plt.show()
```





```
In [24]: # We compute way less points using adaptative timestep
         len(df)
```

```
Out[24]: 168
```

```
In [26]: len(df_005)
```

```
Out[26]: 500
```