## WT fitting (milliseconds)

### WT\_nodb

**Data files**

WTpath =r'C:\Users\delbe\Downloads\wut\wut\FALL\_2018\Post\_grad\accili\2018\lab\digitize\kim2012\new csv\WT\milliseconds'

FApath =r'C:\Users\delbe\Downloads\wut\wut\FALL\_2018\Post\_grad\accili\2018\lab\digitize\kim2012\new csv\FA\milliseconds'

act\_files = glob.glob(path + "/\*act\*.csv")

de\_files = glob.glob(path + "/\*de\*.csv")

ddact = []

aleg = []

for f in act\_files:

prefix = os.path.basename(os.path.normpath(f)) #get the filename without directory/path

if '-1' in str(prefix):

short\_prefix = prefix[-14:-10]

else:

short\_prefix = prefix[-13:-10]

aleg.append(str(short\_prefix))

df = pd.read\_csv(f).rename(columns={'Curve1' : str(short\_prefix)})

ddact.append(df)

dlist = []

for da in ddact:

df = pd.DataFrame.from\_dict(da)

dlist.append(df)

dleg = []

for f in de\_files:

prefix = os.path.basename(os.path.normpath(f)) #get the filename without directory/path

if 'de\_0' in str(prefix):

short\_prefix = 0 #filename

else:

short\_prefix = prefix[-11:-8]

dleg.append(short\_prefix)

df = pd.read\_csv(f).rename(columns={'Curve1' : str(short\_prefix)})

dlist.append(df)

dfMerge = pd.concat(dlist, axis=1, sort=True)

**Initials**

pars\_WT = [9.09177339e-02, 1.29003637e+01, 1.87000004e+03, 9.50000094e+01,

1.25596622e-04, 1.05001820e+01, 7.99616145e-03, 3.80000019e+01,

1.98459327e-03, 2.00000000e-06, 2.05373978e-05, 1.23221776e-02]

**Error**

def rrse(data, model):

n = len(data)

data\_mean = np.mean(data, axis=0)

sum1 = 0

sum2 = 0

for i in range(n):

sum1 += ( data[i] - model[i] )\*\*2

sum2 += ( data\_mean - model[i] )\*\*2

error = sum1/sum2

return np.sqrt(error)

**Bounds**

b\_a0 = [2e-3, 0.5]

b\_sa = [8, 15]

b\_b0 = [500, 1800]

b\_sb = [30, 180]

b\_c0 = [1e-5, 0.1]

b\_sc = [5, 15]

b\_d0 = [7e-3, 1]

b\_sd = [20, 100]

b\_g1 = [3e-4, 3e-2]

b\_h1 = [1e-8, 1e-5]

b\_g2 = [1e-5, 1e-3]

b\_h2 = [1e-3, 1e-1]

**Method**

res = minimize(cost, param, args=(db), method='L-BFGS-B', bounds=bds)

**Result**

fun: 1.6927328557754058

hess\_inv: <12x12 LbfgsInvHessProduct with dtype=float64>

jac: array([ 3.57824792e-01, 7.38424877e-02, 4.37327285e-02, 6.24260643e-03,

2.87257181e+01, -2.20767604e-01, 4.31040657e+01, 3.16621840e-02,

9.41666678e-01, -4.24265922e+03, 9.97440490e+01, -1.22127530e-01])

message: b'CONVERGENCE: REL\_REDUCTION\_OF\_F\_<=\_FACTR\*EPSMCH'

nfev: 1534

nit: 81

status: 0

success: True

x: array([6.33648602e-02, 1.29293559e+01, 1.80000000e+03, 9.49982596e+01,

1.17438409e-04, 1.04735360e+01, 7.00000000e-03, 3.79967432e+01,

2.80975150e-03, 1.00000000e-05, 2.63639353e-05, 1.49911331e-02])

2.1815767545116262 1.6927328557754058

**Output**

015

**Bounds**

(h1)

b\_a0 = [2e-3, 0.5]

b\_sa = [8, 15]

b\_b0 = [500, 1800]

b\_sb = [30, 180]

b\_c0 = [1e-5, 0.1]

b\_sc = [5, 15]

b\_d0 = [7e-3, 1]

b\_sd = [20, 100]

b\_g1 = [3e-4, 3e-2]

**b\_h1 = [1e-5, 1e-3]**

b\_g2 = [1e-5, 1e-3]

b\_h2 = [1e-3, 1e-1]

**Initials**

(015)

6.33648602e-02, 1.29293559e+01, 1.80000000e+03, 9.49982596e+01,

1.17438409e-04, 1.04735360e+01, 7.00000000e-03, 3.79967432e+01,

2.80975150e-03, 1.00000000e-05, 2.63639353e-05, 1.49911331e-02

**Result**

fun: 1.6675351462077281

hess\_inv: <12x12 LbfgsInvHessProduct with dtype=float64>

jac: array([-3.24062783e+00, 1.46457180e-01, 3.66483066e-02, -4.15724344e-02,

1.50317433e+01, -1.80236159e-01, 2.53585649e+01, 1.18727250e-03,

7.68810993e-01, -1.23489713e+01, 6.80566185e+01, -1.83558349e+00])

message: b'CONVERGENCE: REL\_REDUCTION\_OF\_F\_<=\_FACTR\*EPSMCH'

nfev: 494

nit: 30

status: 0

success: True

x: array([6.33917118e-02, 1.29293898e+01, 1.79999977e+03, 9.49982622e+01,

1.35287682e-04, 1.04734438e+01, 7.00000000e-03, 3.79967428e+01,

2.87927806e-03, 2.25579201e-05, 2.25786256e-05, 1.49621738e-02])

1.6927328443990055 1.6675351462077281

**Output**

016

**Initials**

([6.33917118e-02, 1.29293898e+01, 1.79999977e+03, 9.49982622e+01,

1.35287682e-04, 1.04734438e+01, 7.00000000e-03, 3.79967428e+01,

2.87927806e-03, 2.25579201e-05, 2.25786256e-05, 1.49621738e-02])

**Bounds**

b\_a0 = [2e-3, 0.5]

b\_sa = [8, 15]

b\_b0 = [500, 1800]

b\_sb = [30, 180]

b\_c0 = [1e-5, 0.1]

b\_sc = [5, 15]

b\_d0 = [7e-5, 1]

b\_sd = [20, 100]

b\_g1 = [3e-4, 3e-2]

b\_h1 = [1e-5, 1e-3]

b\_g2 = [1e-6, 1e-4]

b\_h2 = [1e-4, 1]

**Method**

res = minimize(cost, param, args=(db), method='L-BFGS-B', bounds=bds)

**Results**

fun: 1.6673498592782314

hess\_inv: <12x12 LbfgsInvHessProduct with dtype=float64>

jac: array([-2.96903278e+00, 1.60814362e-01, -8.61026805e-03, 4.78583617e-02,

-3.37623867e+01, -1.93579108e-01, 3.51559461e+01, -2.84305912e-03,

7.85976233e+00, 1.46376691e+01, 1.05340317e+02, 1.84167221e+00])

message: b'CONVERGENCE: REL\_REDUCTION\_OF\_F\_<=\_FACTR\*EPSMCH'

nfev: 247

nit: 6

status: 0

success: True

x: array([6.34301786e-02, 1.29293755e+01, 1.79999977e+03, 9.49982597e+01,

1.35395346e-04, 1.04734620e+01, 6.99934753e-03, 3.79967430e+01,

2.87905103e-03, 2.24788874e-05, 2.27056799e-05, 1.50489481e-02])

1.6675351459691166 1.6673498592782314

**Output**

20

### WT\_db

**Initials**

6.33917118e-02, 1.29293898e+01, 1.79999977e+03, 9.49982622e+01,

1.04734438e+01, 7.00000000e-03, 3.79967428e+01,

2.87927806e-03, 2.25579201e-05, 2.25786256e-05, 1.49621738e-02

**Method**

res = minimize(cost, param, args=(db), method='L-BFGS-B', bounds=bds)

**Result**

fun: 3.5964545451328025

hess\_inv: <11x11 LbfgsInvHessProduct with dtype=float64>

jac: array([ 1.01471631e+00, 3.27902505e-01, 1.07455289e-01, 1.75521997e-01,

3.11544790e-01, 8.56590354e-02, 2.17383356e-01, 1.71470064e+01,

-1.79420217e+02, 1.17017251e+02, 1.21492567e+00])

message: b'CONVERGENCE: REL\_REDUCTION\_OF\_F\_<=\_FACTR\*EPSMCH'

nfev: 1248

nit: 61

status: 0

success: True

x: array([8.33208013e-02, 1.28221836e+01, 1.79999730e+03, 9.49977019e+01,

1.04750909e+01, 1.15155726e-02, 3.79985221e+01, 2.26920875e-03,

8.32959467e-05, 2.21732367e-05, 1.06000029e-02])

4.456392534418083 3.5964545451328025

**Initials**

N/A

**Bounds**

b\_a0 = [2e-3, 0.5]

b\_sa = [8, 15]

b\_b0 = [500, 1800]

b\_sb = [30, 180]

b\_c0 = [1e-5, 0.1]

b\_sc = [5, 15]

b\_d0 = [7e-5, 1]

b\_sd = [20, 100]

b\_g1 = [3e-4, 3e-2]

b\_h1 = [1e-5, 1e-3]

b\_g2 = [1e-6, 1e-4]

b\_h2 = [1e-4, 1]

**Method**

res = differential\_evolution(cost, args=(db), bounds=bds, maxiter=100)

**Results**

fun: 2.861789800491562

jac: array([-27.03804669, 0.75305011, 0.05338925, 0.24582651,

-0.13099397, 0.80157534, 0.07104926, -0.76722624,

28.24974299, -4.07103435, 1.57308158])

message: 'Maximum number of iterations has been exceeded.'

nfev: 18837

nit: 100

success: False

x: array([1.76103425e-02, 1.27353709e+01, 1.65262977e+03, 3.10382809e+01,

1.23079201e+01, 5.33102518e-03, 9.48152909e+01, 1.66185027e-03,

8.69260949e-05, 1.18421201e-05, 8.35311124e-02])

**Script**

ms\_4s\_WT\_fit\_diffevo.py

**Initials**

[1.76103425e-02, 1.27353709e+01, 1.65262977e+03, 3.10382809e+01,

1.23079201e+01, 5.33102518e-03, 9.48152909e+01, 1.66185027e-03,

8.69260949e-05, 1.18421201e-05, 8.35311124e-02]

**Bounds**

b\_a0 = [2e-3, 0.5]

b\_sa = [8, 15]

b\_b0 = [500, 1800]

b\_sb = [30, 180]

b\_sc = [5, 15]

b\_d0 = [7e-5, 1]

b\_sd = [20, 100]

b\_g1 = [3e-4, 3e-2]

b\_h1 = [1e-5, 1e-3]

b\_g2 = [1e-6, 1e-4]

b\_h2 = [1e-4, 1]

**Method**

res = differential\_evolution(cost, bounds=bds, maxiter=300)

**Results**

fun: 1.9845906063341112

jac: array([-1.57538478e+00, 8.62703464e-02, 2.86344282e-03, 1.75732762e-02,

-2.30691022e-02, 2.36306431e+01, 2.04012363e-03, -1.47153177e+01,

4.08204640e+02, 3.75024596e+02, 1.31031118e+01])

message: 'Maximum number of iterations has been exceeded.'

nfev: 50133

nit: 300

success: False

x: array([1.10867367e-02, 9.55519780e+00, 1.62733571e+03, 5.22360290e+01,

1.22555924e+01, 9.27995454e-03, 6.34796033e+01, 1.10795143e-03,

5.24678017e-05, 1.38252627e-05, 1.36215226e-02])

2.861789789444991 1.9845906063341112

**Initials**

1.10867367e-02, 9.55519780e+00, 1.62733571e+03, 5.22360290e+01,

1.22555924e+01, 9.27995454e-03, 6.34796033e+01, 1.10795143e-03,

5.24678017e-05, 1.38252627e-05, 1.36215226e-02

**Bounds**

b\_a0 = [2e-3, 0.5]

b\_sa = [8, 15]

b\_b0 = [500, 1800]

b\_sb = [30, 180]

b\_c0 = [1e-5, 0.1]

b\_sc = [5, 15]

b\_d0 = [7e-5, 1]

b\_sd = [20, 100]

b\_g1 = [3e-4, 3e-2]

b\_h1 = [1e-5, 1e-3]

b\_g2 = [1e-6, 1e-4]

b\_h2 = [1e-4, 1]

**Method**

res = differential\_evolution(cost, bounds=bds, maxiter=500)

**Results**

fun: 1.9828619538194605

jac: array([-6.95471014e-02, 3.97314404e-02, -1.02764464e-03, 5.86497517e-03,

-1.27155841e-02, 7.36790451e-01, -3.41748851e-03, -1.21738117e+00,

2.12952629e+02, -4.06054119e+00, -1.05931490e+00])

message: 'Optimization terminated successfully.'

nfev: 75696

nit: 455

success: True

x: array([1.13366555e-02, 1.00180160e+01, 1.42142648e+03, 4.48005383e+01,

1.31550294e+01, 9.67525119e-03, 5.81245359e+01, 1.11104011e-03,

5.16354448e-05, 1.31609309e-05, 1.26205363e-02])

1.9845906066397632 1.9828619538194605

**Output**

019

## FA fitting (milliseconds)

### FA\_db

**Initials**

1.45593110e-02, 1.12943566e+01, 7.80034122e+02, 7.99236433e+01,

1.12888905e+01, 5.25844323e-02, 7.99634857e+01,

2.80222077e-03, 2.73683443e-04, 5.70755258e-05, 5.55860814e-03

**Method**

res = minimize(cost, param, args=(db), method='L-BFGS-B', bounds=bds)

**Results**

fun: 4.803968650566543

hess\_inv: <11x11 LbfgsInvHessProduct with dtype=float64>

jac: array([ 3.85293442e+01, -1.74823711e-01, -1.80012449e-02, -1.77157844e-02,

7.68383579e-02, 8.19139423e-02, 1.18959065e-02, 7.09675905e+01,

-1.13224165e+03, -6.02232443e+02, 9.23761618e+02])

message: b'CONVERGENCE: REL\_REDUCTION\_OF\_F\_<=\_FACTR\*EPSMCH'

nfev: 1704

nit: 73

status: 0

success: True

x: array([3.24125675e-03, 1.00661350e+01, 7.80022321e+02, 7.98762728e+01,

9.82740261e+00, 7.46522524e-02, 7.99537785e+01, 1.69239743e-03,

2.20433077e-05, 1.12195412e-05, 1.00000000e-03])

11.233630147867906 4.803968650566543

**Bounds**

b\_a0 = [2e-3, 0.5]

b\_sa = [8, 15]

b\_b0 = [500, 1800]

b\_sb = [30, 180]

b\_c0 = [1e-5, 0.1]

b\_sc = [5, 15]

b\_d0 = [7e-3, 1]

b\_sd = [20, 100]

b\_g1 = [3e-4, 3e-2]

b\_h1 = [1e-6, 1e-4]

b\_g2 = [1e-6, 1e-4]

b\_h2 = [1e-5, 1e-3]

**Initials**

(above)

3.24125675e-03, 1.00661350e+01, 7.80022321e+02, 7.98762728e+01,

9.82740261e+00, 7.46522524e-02, 7.99537785e+01, 1.69239743e-03,

2.20433077e-05, 1.12195412e-05, 1.00000000e-03

**Method**

res = minimize(cost, param, args=(db), method='L-BFGS-B', bounds=bds)

**Result**

fun: 2.32316607107683

hess\_inv: <11x11 LbfgsInvHessProduct with dtype=float64>

jac: array([ 2.15248366e+00, -5.97262240e-02, -4.77089479e-03, -6.40993925e-03,

6.05326012e-02, 5.30779420e-02, -1.61888281e-03, 3.14313522e+01,

-9.08850339e+02, 4.44332516e+03, -3.26033036e+01])

message: b'CONVERGENCE: REL\_REDUCTION\_OF\_F\_<=\_FACTR\*EPSMCH'

nfev: 756

nit: 30

status: 0

success: True

x: array([3.36505155e-03, 1.00347218e+01, 7.80026840e+02, 7.98755923e+01,

9.82907636e+00, 8.60554543e-02, 7.99439728e+01, 1.61649730e-03,

2.25588426e-05, 7.51177790e-06, 2.53662197e-04])

4.803968643328568 2.32316607107683

**Initials**

3.36505155e-03, 1.00347218e+01, 7.80026840e+02, 7.98755923e+01,

9.82907636e+00, 8.60554543e-02, 7.99439728e+01, 1.61649730e-03,

2.25588426e-05, 7.51177790e-06, 2.53662197e-04

**Method**

#basinhop

minimizer\_kwargs = dict(args=(db), method="L-BFGS-B", bounds=bds)

res = basinhopping(cost, param, niter=100, minimizer\_kwargs=minimizer\_kwargs)

**Result**

fun: 1.9841524154033934

lowest\_optimization\_result: fun: 1.9841524154033934

hess\_inv: <11x11 LbfgsInvHessProduct with dtype=float64>

jac: array([ 5.96695267e+00, -8.38885850e-02, 1.88872473e-02, 1.74971593e-02,

2.07538209e-02, 6.41847029e+00, -5.62914160e-03, -5.45510431e+01,

2.08129315e+03, -1.09449760e+03, -1.06992960e+02])

message: b'CONVERGENCE: REL\_REDUCTION\_OF\_F\_<=\_FACTR\*EPSMCH'

nfev: 2112

nit: 97

status: 0

success: True

x: array([1.26564530e-02, 1.11020326e+01, 7.76891719e+02, 9.45273368e+01,

6.79750530e+00, 7.00000000e-03, 7.75018638e+01, 1.51109319e-03,

9.32336062e-05, 6.54519949e-06, 2.67686812e-04])

message: ['requested number of basinhopping iterations completed successfully']

minimization\_failures: 13

nfev: 107208

nit: 100

x: array([1.26564530e-02, 1.11020326e+01, 7.76891719e+02, 9.45273368e+01,

6.79750530e+00, 7.00000000e-03, 7.75018638e+01, 1.51109319e-03,

9.32336062e-05, 6.54519949e-06, 2.67686812e-04])

2.3231660686706905 1.9841524154033934

**Output**

017

Same initials and method as above

**Results**

fun: 1.9794677518320212

lowest\_optimization\_result: fun: 1.9794677518320212

hess\_inv: <11x11 LbfgsInvHessProduct with dtype=float64>

jac: array([ 2.87258193e+00, -4.78505680e-02, -5.90003602e-03, -1.07517106e-02,

6.79083456e-03, 6.52907075e+00, 7.09121650e-04, 2.34712688e+01,

-2.44738734e+02, 4.12424878e+03, -1.49194046e+01])

message: b'CONVERGENCE: REL\_REDUCTION\_OF\_F\_<=\_FACTR\*EPSMCH'

nfev: 924

nit: 40

status: 0

success: True

x: array([1.15654950e-02, 1.18827173e+01, 7.72477305e+02, 5.95362467e+01,

7.22673947e+00, 7.00000000e-03, 6.64662684e+01, 1.56868968e-03,

8.80185239e-05, 6.55974192e-06, 2.66435453e-04])

message: ['requested number of basinhopping iterations completed successfully']

minimization\_failures: 12

nfev: 96996

nit: 100

x: array([1.15654950e-02, 1.18827173e+01, 7.72477305e+02, 5.95362467e+01,

7.22673947e+00, 7.00000000e-03, 6.64662684e+01, 1.56868968e-03,

8.80185239e-05, 6.55974192e-06, 2.66435453e-04])

2.3231660686706905 1.9794677518320212

**Output**

018

### FA\_nodb

**Initials**

[3.36505155e-03, 1.00347218e+01, 7.80026840e+02, 7.98755923e+01,

1e-2, 9.82907636e+00, 8.60554543e-02, 7.99439728e+01, 1.61649730e-03,

2.25588426e-05, 7.51177790e-06, 2.53662197e-04]

**Method**

Basinhop x 100

**Results**

fun: 2.1088698180187104

lowest\_optimization\_result: fun: 2.1088698180187104

hess\_inv: <12x12 LbfgsInvHessProduct with dtype=float64>

jac: array([-1.55837911e+00, 1.04069731e-01, 4.65356198e-02, 3.16257687e-02,

3.67782382e-01, 3.82042398e-02, 5.31449995e-02, 3.99505318e-02,

-1.22488212e+01, 1.41906495e+03, 1.28305179e+03, -4.67007276e+01])

message: b'CONVERGENCE: REL\_REDUCTION\_OF\_F\_<=\_FACTR\*EPSMCH'

nfev: 1599

nit: 75

status: 0

success: True

x: array([1.31501601e-02, 1.17453736e+01, 7.81312947e+02, 7.62266428e+01,

6.80571170e-02, 1.12548126e+01, 9.99422854e-01, 7.87130095e+01,

1.69460949e-03, 8.68347267e-05, 6.65905483e-06, 2.71883161e-04])

message: ['requested number of basinhopping iterations completed successfully']

minimization\_failures: 10

nfev: 119665

nit: 100

x: array([1.31501601e-02, 1.17453736e+01, 7.81312947e+02, 7.62266428e+01,

6.80571170e-02, 1.12548126e+01, 9.99422854e-01, 7.87130095e+01,

1.69460949e-03, 8.68347267e-05, 6.65905483e-06, 2.71883161e-04])

3.975154332903597 2.1088698180187104

**Method**

res = differential\_evolution(cost, bounds=bds, maxiter=500)

**Results**

fun: 2.0288909952352165

jac: array([ 1.81941311e+00, -3.45636408e-02, -2.07101891e-02, -3.34415606e-02,

-1.49967239e-01, -7.11568582e-03, 7.10085324e-03, -2.42728504e-02,

1.89205452e+01, -1.34082654e+03, 3.34695391e+03, 1.24699432e+03])

message: 'Optimization terminated successfully.'

nfev: 41894

nit: 229

success: True

x: array([2.81392598e-02, 1.24321776e+01, 1.69887431e+03, 5.26916066e+01,

1.02940638e-02, 7.51102677e+00, 8.20919105e-01, 4.10800310e+01,

1.61156635e-03, 9.00601376e-05, 6.35660215e-06, 2.64059710e-04])

3.975154332903597 2.0288909952352165

**Initials**

2.81392598e-02, 1.24321776e+01, 1.69887431e+03, 5.26916066e+01,

1.02940638e-02, 7.51102677e+00, 8.20919105e-01, 4.10800310e+01,

1.61156635e-03, 9.00601376e-05, 6.35660215e-06, 2.64059710e-04

**Bounds**

b\_a0 = [2e-3, 0.5]

b\_sa = [8, 15]

b\_b0 = [500, 1800]

b\_sb = [30, 180]

b\_c0 = [1e-5, 0.1]

b\_sc = [5, 15]

b\_d0 = [7e-3, 1]

b\_sd = [20, 100]

b\_g1 = [3e-4, 3e-2]

b\_h1 = [1e-6, 1e-4]

b\_g2 = [1e-6, 1e-4]

b\_h2 = [1e-5, 1e-3]

**Method**

res = minimize(cost, param, args=(db), method='L-BFGS-B', bounds=bds)

**Results**

fun: 2.028455989467702

hess\_inv: <12x12 LbfgsInvHessProduct with dtype=float64>

jac: array([-5.04704469e+00, 1.25116051e-01, 9.81468240e-03, 8.14217582e-02,

2.89122193e-01, 3.36022765e-02, 3.57587293e-02, 1.96726635e-02,

-1.03787776e+02, -5.02004080e+02, -1.94910109e+02, 1.03322452e+03])

message: b'CONVERGENCE: REL\_REDUCTION\_OF\_F\_<=\_FACTR\*EPSMCH'

nfev: 819

nit: 7

status: 0

success: True

x: array([2.81123979e-02, 1.24321539e+01, 1.69887433e+03, 5.26916050e+01,

1.03830274e-02, 7.51102369e+00, 8.20907957e-01, 4.10800331e+01,

1.61037396e-03, 9.00771840e-05, 6.35853372e-06, 2.63637305e-04])

2.0288909942546867 2.028455989467702

**Output**

021

## Final

**FA\_db0** [0.011565495, 11.8827173, 772.477305, 59.5362467, 7.58655751788942e-05, 7.22673947, 0.007, 66.4662684, 0.00156868968, 8.80185239e-05, 6.55974192e-06, 0.000266435453]

RRSE = 1.9794677518320212

18

**WT\_db0** [0.0113366555, 10.018016, 1421.42648, 44.8005383, 0.0015921933784909365, 13.1550294, 0.00967525119, 58.1245359, 0.00111104011, 5.16354448e-05, 1.31609309e-05, 0.0126205363]

RRSE = 1.9828619538194605

19

**FA\_nodb** [0.0281123979, 12.4321539, 1698.87433, 52.691605, 0.0103830274, 7.51102369, 0.820907957, 41.0800331, 0.00161037396, 9.0077184e-05, 6.35853372e-06, 0.000263637305]

RRSE = 2.028455989467702

21

**WT\_nodb** [0.0634301786, 12.9293755, 1799.99977, 94.9982597, 0.000135395346, 10.473462, 0.00699934753, 37.996743, 0.00287905103, 2.24788874e-05, 2.27056799e-05, 0.0150489481]

RRSE = 1.6673498592782314

20

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **WT\_nodb** | **FA\_nodb** | **WT\_db0** | **FA\_db0** |
| **α0** | 0.06343 | 0.028112 | 0.011337 | 0.011565 |
| **sα** | 12.92938 | 12.43215 | 10.01802 | 11.88272 |
| **β0** | 1800 | 1698.874 | 1421.426 | 772.4773 |
| **sβ** | 94.99826 | 52.69161 | 44.80054 | 59.53625 |
| **α’0** | 0.000135 | 0.010383 | 0.001592 | 7.59E-05 |
| **sα’** | 10.47346 | 7.511024 | 13.15503 | 7.226739 |
| **β’0** | 0.006999 | 0.820908 | 0.009675 | 0.007 |
| **sβ’** | 37.99674 | 41.08003 | 58.12454 | 66.46627 |
| **g** | 0.002879 | 0.00161 | 0.001111 | 0.001569 |
| **h** | 2.25E-05 | 9.01E-05 | 5.16E-05 | 8.8E-05 |
| **g'** | 2.27E-05 | 6.36E-06 | 1.32E-05 | 6.56E-06 |
| **h'** | 0.015049 | 0.000264 | 0.012621 | 0.000266 |

**Table 1. Parameter sets**

### Compare

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **FA/WT** | | **Nodb/db\_0** | |
|  | **Nodb** | **Db\_0** | **WT/WT** | **FA/FA** |
| **α0** | **0.443202** | 1.020186 | **5.59514** | **2.430713** |
| **sα** | 0.961543 | 1.186135 | 1.290612 | 1.046238 |
| **β0** | 0.943819 | **0.543452** | 1.266333 | **2.199255** |
| **sβ** | **0.554659** | 1.328918 | **2.120471** | 0.885034 |
| **α’0** | **76.68674** | **0.047648** | **0.085037** | **136.8609** |
| **sα’** | 0.717148 | **0.549352** | 0.796156 | 1.039338 |
| **β’0** | **117.2835** | 0.723495 | 0.723428 | **117.2726** |
| **sβ’** | 1.081146 | 1.143515 | 0.653713 | 0.618058 |
| **g** | **0.559342** | 1.411911 | **2.591312** | 1.026573 |
| **h** | **4.00719** | **1.704614** | **0.435338** | 1.023389 |
| **g'** | **0.280042** | **0.498425** | 1.725234 | 0.969327 |
| **h'** | **0.017519** | **0.021111** | 1.192417 | 0.989498 |

**Table 2. Ratios of parameters between indicated parameter sets**

Red: ratio < 0.5, Blue: ratio > 2, Black: within 0.05 of either 0.5 or 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **FA/WT** | | **Nodb/db\_0** | |
|  | **Nodb** | **db\_0** | **WT/WT** | **FA/FA** |
| **0** | d0 | **h2** | c0 | c0 |
| **1** | c0 | c0 | a0 | d0 |
| **2** | **h2** | **g2** | **g1** | a0 |
| **3** | **h1** | b0 | **h1** | b0 |
| **4** | **g2** | sc | sb | sd |
| **5** | a0 | **h1** | **g2** | sb |
| **6** | sb | **g1** | sd | sa |
| **7** | **g1** | d0 | d0 | sc |
| **8** | sc | sb | sa | **g2** |
| **9** | sd | sa | b0 | **g1** |
| **10** | b0 | sd | sc | **h1** |
| **11** | sa | a0 | **h2** | **h2** |

**Table 3. Most changed parameters in each ratio set in decreasing order.**

Bold: voltage-independent rate constants

### Arrow

## Constrained Fits of FA\_db0

Basinhopping x100

### H2

[2.99761407]

fun: array([2.99761407])

lowest\_optimization\_result: fun: array([2.99761407])

hess\_inv: <1x1 LbfgsInvHessProduct with dtype=float64>

jac: array([0.0584889])

message: b'CONVERGENCE: REL\_REDUCTION\_OF\_F\_<=\_FACTR\*EPSMCH'

nfev: 30

nit: 9

status: 0

success: True

x: array([0.00026328])

message: ['requested number of basinhopping iterations completed successfully']

minimization\_failures: 0

nfev: 2256

nit: 100

x: array([0.00026328])

### G2, H2

fun: 2.851149504143115

lowest\_optimization\_result: fun: 2.851149504143115

hess\_inv: <2x2 LbfgsInvHessProduct with dtype=float64>

jac: array([7.92469548, 2.21634724])

message: b'CONVERGENCE: REL\_REDUCTION\_OF\_F\_<=\_FACTR\*EPSMCH'

nfev: 105

nit: 11

status: 0

success: True

x: array([8.56004280e-06, 2.32102942e-04])

message: ['requested number of basinhopping iterations completed successfully']

minimization\_failures: 0

nfev: 6822

nit: 100

x: array([8.56004280e-06, 2.32102942e-04])

3.033170280444241 2.851149504143115

### B0, g2, h2

fun: 3.48354825468558

lowest\_optimization\_result: fun: 3.48354825468558

hess\_inv: <3x3 LbfgsInvHessProduct with dtype=float64>

jac: array([-1.89075422e-03, 2.39994541e+02, -8.08032663e+00])

message: b'CONVERGENCE: REL\_REDUCTION\_OF\_F\_<=\_FACTR\*EPSMCH'

nfev: 152

nit: 13

status: 0

success: True

x: array([7.83511574e+02, 5.23887295e-06, 2.34056521e-04])

message: ['requested number of basinhopping iterations completed successfully']

minimization\_failures: 1

nfev: 13320

nit: 100

x: array([7.83511574e+02, 5.23887295e-06, 2.34056521e-04])

3.591125718183997 3.48354825468558

### Sc, g1, h1, g2, h2

fun: 2.181508094648328

lowest\_optimization\_result: fun: 2.181508094648328

hess\_inv: <5x5 LbfgsInvHessProduct with dtype=float64>

jac: array([ 2.74629208e-03, -6.43563864e+00, 8.61179172e+01, -1.51095725e+03,

5.18737084e+00])

message: b'CONVERGENCE: REL\_REDUCTION\_OF\_F\_<=\_FACTR\*EPSMCH'

nfev: 846

nit: 36

status: 0

success: True

x: array([5.40158955e+00, 7.15183837e-04, 9.86811021e-05, 6.80569957e-06,

2.42970157e-04])

message: ['requested number of basinhopping iterations completed successfully']

minimization\_failures: 1

nfev: 25932

nit: 50

x: array([5.40158955e+00, 7.15183837e-04, 9.86811021e-05, 6.80569957e-06,

2.42970157e-04])

3.6167944384652717 2.181508094648328

### B0, g1, h1, g2, h2

fun: 2.4904779858426265

lowest\_optimization\_result: fun: 2.4904779858426265

hess\_inv: <5x5 LbfgsInvHessProduct with dtype=float64>

jac: array([ 3.56621399e-03, 4.38008163e-01, -4.26060121e-01, 1.16562462e+02,

-7.41506687e+00])

message: b'CONVERGENCE: REL\_REDUCTION\_OF\_F\_<=\_FACTR\*EPSMCH'

nfev: 846

nit: 47

status: 0

success: True

x: array([1.46128968e+03, 7.79550287e-04, 9.06739152e-05, 1.18360859e-05,

2.30109864e-04])

message: ['requested number of basinhopping iterations completed successfully']

minimization\_failures: 1

nfev: 47832

nit: 100

x: array([1.46128968e+03, 7.79550287e-04, 9.06739152e-05, 1.18360859e-05,

2.30109864e-04])

4.205504428859608 2.4904779858426265

### Sc, b0, g1, g2, h1, h2

fun: 2.007017112680137

lowest\_optimization\_result: fun: 2.007017112680137

hess\_inv: <6x6 LbfgsInvHessProduct with dtype=float64>

jac: array([ 2.88147284e-03, 9.68913838e-04, -1.83065453e+01, 5.90104999e+01,

-1.51575239e+03, 1.97147916e+00])

message: b'CONVERGENCE: REL\_REDUCTION\_OF\_F\_<=\_FACTR\*EPSMCH'

nfev: 1029

nit: 75

status: 0

success: True

x: array([1.79480105e+03, 5.61637526e+00, 8.14314714e-04, 8.39878730e-05,

6.79288196e-06, 2.41754931e-04])

message: ['requested number of basinhopping iterations completed successfully']

minimization\_failures: 0

nfev: 63014

nit: 100

x: array([1.79480105e+03, 5.61637526e+00, 8.14314714e-04, 8.39878730e-05,

6.79288196e-06, 2.41754931e-04])

4.529398748614678 2.007017112680137

### D0, sc, b0, g1, g2, h1, h2

### G1, g2, h1, h2

fun: 2.5002131409668675

lowest\_optimization\_result: fun: 2.5002131409668675

hess\_inv: <4x4 LbfgsInvHessProduct with dtype=float64>

jac: array([ -2.18548357, -13.40792068, 83.02173669, 3.26934453])

message: b'CONVERGENCE: REL\_REDUCTION\_OF\_F\_<=\_FACTR\*EPSMCH'

nfev: 240

nit: 26

status: 0

success: True

x: array([7.68752772e-04, 1.00000000e-04, 1.17472275e-05, 2.29815551e-04])

message: ['requested number of basinhopping iterations completed successfully']

minimization\_failures: 0

nfev: 29020

nit: 100

x: array([7.68752772e-04, 1.00000000e-04, 1.17472275e-05, 2.29815551e-04])

5.560584253076787 2.5002131409668675

fun: 2.5002054874990147

lowest\_optimization\_result: fun: 2.5002054874990147

hess\_inv: <4x4 LbfgsInvHessProduct with dtype=float64>

jac: array([ 2.07786717, 0.64572121, 194.60539624, -5.04445108])

message: b'CONVERGENCE: REL\_REDUCTION\_OF\_F\_<=\_FACTR\*EPSMCH'

nfev: 560

nit: 45

status: 0

success: True

x: array([7.69143105e-04, 1.01265062e-04, 1.17413419e-05, 2.29690294e-04])

message: ['requested number of basinhopping iterations completed successfully']

minimization\_failures: 0

nfev: 73675

nit: 200

x: array([7.69143105e-04, 1.01265062e-04, 1.17413419e-05, 2.29690294e-04])

3.60133672640402 2.5002054874990147

## Constrained Fits of FA\_nodb

### H2

[4.39108814]

fun: array([4.39108814])

lowest\_optimization\_result: fun: array([4.39108814])

hess\_inv: <1x1 LbfgsInvHessProduct with dtype=float64>

jac: array([0.010135])

message: b'CONVERGENCE: REL\_REDUCTION\_OF\_F\_<=\_FACTR\*EPSMCH'

nfev: 32

nit: 8

status: 0

success: True

x: array([0.00028551])

message: ['requested number of basinhopping iterations completed successfully']

minimization\_failures: 0

nfev: 2522

nit: 100

x: array([0.00028551])

### G2, H2

fun: 2.9205522224916844

lowest\_optimization\_result: fun: 2.9205522224916844

hess\_inv: <2x2 LbfgsInvHessProduct with dtype=float64>

jac: array([446.6050012 , 4.90657737])

message: b'CONVERGENCE: REL\_REDUCTION\_OF\_F\_<=\_FACTR\*EPSMCH'

nfev: 51

nit: 6

status: 0

success: True

x: array([1.81137799e-06, 2.33446086e-04])

message: ['requested number of basinhopping iterations completed successfully']

minimization\_failures: 24

nfev: 13305

nit: 100

x: array([1.81137799e-06, 2.33446086e-04])

3.3984463654336405 2.9205522224916844

### G1, h1, g2, h2

fun: 1.893895529144348

lowest\_optimization\_result: fun: 1.893895529144348

hess\_inv: <4x4 LbfgsInvHessProduct with dtype=float64>

jac: array([ 5.11541431e-01, -2.35147103e+03, 4.79638218e+02, -9.23042132e-01])

message: b'CONVERGENCE: REL\_REDUCTION\_OF\_F\_<=\_FACTR\*EPSMCH'

nfev: 425

nit: 36

status: 0

success: True

x: array([2.03468968e-03, 1.00000000e-04, 5.26398329e-06, 2.36159399e-04])

message: ['requested number of basinhopping iterations completed successfully']

minimization\_failures: 4

nfev: 25585

nit: 100

x: array([2.03468968e-03, 1.00000000e-04, 5.26398329e-06, 2.36159399e-04])

5.299361074109219 1.893895529144348

fun: 1.893894934789574

lowest\_optimization\_result: fun: 1.893894934789574

hess\_inv: <4x4 LbfgsInvHessProduct with dtype=float64>

jac: array([ 1.42687617e-01, -2.35691216e+03, 7.42286234e+02, -2.21055863e-01])

message: b'CONVERGENCE: REL\_REDUCTION\_OF\_F\_<=\_FACTR\*EPSMCH'

nfev: 360

nit: 27

status: 0

success: True

x: array([2.03431899e-03, 1.00000000e-04, 5.26726209e-06, 2.36210348e-04])

message: ['requested number of basinhopping iterations completed successfully']

minimization\_failures: 53

nfev: 276060

nit: 1000

x: array([2.03431899e-03, 1.00000000e-04, 5.26726209e-06, 2.36210348e-04])

2.2545223711207423 1.893894934789574

## Revised detailed balance fitting in WT

### A0

00

[15.765366298369525, 12.9293755, 1799.99977, 94.9982597, 0.020937615801900906, 10.473462, 0.00699934753, 37.996743, 0.00287905103, 2.24788874e-05, 2.27056799e-05, 0.0150489481]

01

Sa = 32

param [15.765366298369525, 32, 1799.99977, 94.9982597, 0.020937615801900906, 10.473462, 0.00699934753, 37.996743, 0.00287905103, 2.24788874e-05, 2.27056799e-05, 0.0150489481]

### B0

00

param [0.0634301786, 12.9293755, 0.0005555556265432189, 94.9982597, 0.020937615801900906, 10.473462, 0.00699934753, 37.996743, 0.00287905103, 2.24788874e-05, 2.27056799e-05, 0.0150489481]

01

Sa = 1

### D0

00

param [0.0634301786, 12.9293755, 1799.99977, 94.9982597, 0.020937615801900906, 10.473462, 0.006999347530000001, 37.996743, 0.00287905103, 2.24788874e-05, 2.27056799e-05, 0.0150489481]

### c0

00

Using db0 params (see above):

* param[1] = 10.9
* param[3] = 85
* param[5] = 15.8
* param[7] = 43