```
//This procedure will take a set of currents and corresponding pressure inputs and do an iterative fit to a 3 state model, minimizing the residual. This procedure will work with raw data collected at 5 kHz. In this procedure, rate d is always held as dependent on the other rates to obey microscopic reversibility.

//To run this procedure, waves need to be pre-loaded for the pressure input (named '"pressure"+frqstr', where frqstr is a number (i.e., frequency)) and for the real data (assumed that it is normalized to a test pulse, truncated to start at the onset of a 4 s stimulus, and named '"avg"+frqstr').

Function VFR_d_ifromo(ra,rb,rc,rd,re,rf,rg,rh,slope,delay,name)

variable ra //closed to open; pressure dependent
variable rb //open to closed
variable rc //open to inact1
variable rd //inact1 to open
variable re //inact1 to closed; pressure dependent
variable rf //closed to inactivated
variable rg //open to inact2
variable rh //inact2 to open
variable delay //offset between pressure clamp and response
variable slope //pressure dependent slope factor; negative for a, positive for e
//one must put in initial guesses for each rate constant; constants all should be in terms of ms-1 except for slope, which should be in units of mmHg

string name //unique name for this run of the fitting routine

//this section creates the waves needed

Make/O/N=20000 Popen = 0 //probability of channel being open
Make/O/N=20000 Pinact = 0 //probability of channel being in inact1
Make/O/N=20000 Pinact2 = 0 // probability of channel being in inact2
Make/O/N = 20000 Pclosed = 0 //probability of channel being closed
Make/O/N = 20000 RaPressure = 0//updates the value of a based on the pressure input
Make/O/N = 20000 RePressure = 0//updates the value of e based on the pressure input
Make/O/N = 20000 Timewave = 0// xaxis for final waves; note that everything is based on sampling of 5Khz
Make/O/N=20000 Popendif = 0// used later to calculate the residual
Make/O/N= 15000 Timewavetest = 0// xaxis for the test pulse, which is sampled at 50kHz; everything else as above
Make/O/N= 15000 PressureTest = 0
Make/O/N =15000 PopenTest = 0
Make/O/N=15000 PInactTest = 0
Make/O/N=15000 PInactTest2 = 0
Make/O/N=15000 PclosedTest = 0
Make/O/N=15000 RaPTest = 0
Make/O/N=15000 RePTest = 0
Make/O/N=21 Rawave  = 0 //this will store all the rate "A"s checked for each iteration
Make/O/N=21 Rbwave = 0
Make/O/N=21 Rcwave = 0
Make/O/N=21 Rdwave = 0
Make/O/N=21 Rewave = 0
Make/O/N=21 Rfwave = 0
Make/O/N=21 Rgwave = 0
Make/O/N=21 Rhwave = 0
Make/O/N=21 slopewave = 0
Make/O/N=21 delaywave = 0


Make/O/N=1000 store_a  = 0//stores the values of rate a for each round of fitting (capped at 100)
```

```
Make/O/N=1000 store_b = 0
Make/O/N=1000 store_c = 0
Make/O/N=1000 store_d = 0
Make/O/N=1000 store_e = 0
Make/O/N=1000 store_f = 0
Make/O/N=1000 store_g = 0
Make/O/N=1000 store_h = 0
Make/O/N=1000 store_slope = 0
Make/O/N=1000 store_delay = 0
Make/O/N=1000 store_residual = 0 //stores the residual for each round, for keeping track of when it is minimized

ra = ra/5; rb = rb/5; rc = rc/5; rd = rd/5; re = re/5; rf = rf/5; rg = rg/5; rh = rh/5 //calibrates constants for 5 khz sampling (assuming they are currently in /ms)

rd = ra*rc*re/(rf*rb) //insures that the model initially obeys microscopic reversibility

Make/O/N=4 Freqwave //this is the wave of frequencies you want to check; if you add/subtract, update N for freqwave and the index2 check at the end. Could also be adapted to test any given number of unique pressure inputs, does not necessarily need to be sinusoidal and/or frequency dependent
Freqwave={2,5,10,20}  //for this run, frequencies of 2, 5, 10, and 20 will be checked
variable index2 = 0 //index that cycles through the pressure input waves used (here, the 4 frequencies)
string frqstr = num2str(freqwave[index2]) //names the output waves with the frequency input

Make/O/N=4 residual //temporarily stores the residual between real and simulated data for each frequency
Make/O/N=21 checker //temporarily stores the average residual for all frequencies tested for each of the 21 values checked for a for a given rate constant in that run

Do //makes graphs for each frequency and overlays them with the real data to visually inspect fit
      Duplicate/O Popen $("Popen_"+frqstr)
      display $("Popen_"+frqstr), $("Avg_"+frqstr) vs timewave
      modifygraph rgb($("Popen_"+frqstr))=(0,0,0)
      index2 += 1
      frqstr = num2str(freqwave[index2])
while (index2 <= 3)



variable index3 = 0  //this controls the 21 values checked for each constant
variable index4 = 0  //this controls iterations (caps at 1000 times one full run through all rate constants, so the program won't run indefinitely)
variable scalefactor = 0 //this is what will be calculated from the test pulse to scale the Popen data

//starting with rate a, increasing/decreasing by 1%

do //this loop will go through 1000 iterations of checking 21 values for each rate constant

            rawave =
{ra*.99,ra*.991,ra*.992,ra*.993,ra*.994,ra*.995,ra*.996,ra*.997,ra*.998,ra*.999,ra,ra*1.001,ra*1.002,ra*1.003,ra*
1.004,ra*1.005,ra*1.006,ra*1.007,ra*1.008,ra*1.009,ra*1.01}
      checker = 0;  residual = 0;  index3 = 0
      rd = rawave [index3]*rc*re/(rb*rf) //updates d for microscopic reversibility

      do   //this loop will check the 21 values for rate a

            index2 = 0 //cycles through frequencies
            frqstr = num2str(freqwave[index2])
```

```
Popen = 0; Pinact = 0; Pinact2 = 0; Pclosed = 0;  residual = 0

do //this loop cycles through the 4 pressures
    variable index = 0 // index for each point of the wave (20000 for a 4 s stimulus at 5 kHz)
    Popen = 0; Pinact = 0; Pinact2 = 0; Pclosed = 0
    Timewave[index] = 0
    duplicate $("pressure"+frqstr) Pressure //temporarily renames real pressure
    Pressure = Pressure*50 //converts the pressure input to mmHg (conversion is 20 mV/1 mmHg)
    Popen[index,index+delay]=0;   Pinact[index,index+delay]=0; Pinact2[index,index+delay]=0
    Pclosed[index,index+delay]=1 //initiated with all channels closed

    Do //this loop will simulate currents for one pressure
        index +=1
        Timewave[index] = index*.0002 //again, assuming a 5 kHz sampling rate
        raPressure[index+delay]= rawave[index3]*exp(pressure[index]/slope*-1) //this is the only rate
constant being varied per iteration here (controlled by index3); a also has to be updated based on the given pressure
        rePressure[index+delay] = re*exp(pressure[index]/slope) //rate e also has to be updated for each
sampling point based on the current pressure
        Popen[index+delay] = Popen[(index+delay-1)] + Pclosed[(index+delay-
1)]*(raPressure[index+delay]) + Pinact[(index+delay-1)]*rd - Popen[(index+delay-1)]*(rb+rc+ rg)+
Pinact2[(index+delay-1)]*rh
        Pinact[index+delay] = PInact[(index+delay-1)] + Popen[(index+delay-1)]*rc +
Pclosed[(index+delay-1)]*rf - Pinact[(index+delay-1)]*(rd+rePressure[index+delay])
        Pinact2[index+delay] = PInact2[index+delay-1] + Popen[(index+delay-1)]*rg -
Pinact2[(index+delay-1)]*rh
        Pclosed[index+delay] = Pclosed[(index+delay-1)] + PInact[(index+delay-
1)]*rePressure[index+delay] + Popen[(index+delay-1)]*rb - Pclosed[(index+delay-1)]*(raPressure[index+delay]+rf)
    while (index <= 20000)

    killwaves pressure //gets rid of the temporary pressure input

    //now making testpulse wave

    index = 0 // index to create various Po waves (300 ms at 50 kHz)
    Pressuretest[index] = -50 //here the test pulse is just a direct step from 0 to -50 mmHg
    Popentest = 0; Pinacttest = 0; Pclosedtest = 0; Popentest[index] = 0; Pinacttest[index] = 0;
    Pinacttest2[index]=0
    Pclosedtest[index] = 1 //again setting all channels initially to closed
    Timewavetest[index] = 0

    Do //this loop will make a test pulse
        index +=1
        Timewavetest[index] = index*.00002 //50 kHz sampling here to avoid oscillations; note that all
rate constants are multiplied by 0.1 to account for this
        Pressuretest[index] = -50
        raPtest[index]= rawave[index3]*.1*exp(pressuretest[index]/slope*-1)
        rePtest[index] = re*.1*exp(pressuretest[index]/slope)
        Popentest[index] = Popentest[(index-1)] + Pclosedtest[(index-1)]*(raPtest[index]) +
Pinacttest[(index-1)]*rd*.1 - Popentest[(index-1)]*(rb*.1+rc*.1+rg*.1) + Pinacttest2[(index-1)]*rh*.1
        Pinacttest[index] = PInacttest[(index-1)] + Popentest[(index-1)]*rc*.1 + Pclosedtest[(index-
1)]*rf*.1 - Pinacttest[(index-1)]*(rd*.1+rePtest[index])
        Pinacttest2[index] = Pinacttest2[(index-1)] + Popentest[(index-1)]*rg*.1 - Pinacttest2[(index-
1)]*rh*.1
        Pclosedtest[index] = Pclosedtest[(index-1)] + PInacttest[(index-1)]*rePtest[index] +
Popentest[(index-1)]*rb*.1 - Pclosedtest[(index-1)]*(raPtest[index]+rf*.1)
    while (index <= 15000)
```

```
            wavestats/q Popentest
            scalefactor = V_max //finds the “peak” of the test pulse and uses it as the scale factor (similar to how the input real data were normalized)

            Duplicate/O $("avg_"+frqstr) realdata //temporarily renames the input real data
            PopenDif = abs((Popen/scalefactor) - (realdata)) //calculates difference between real and simulated data
            killwaves realdata
            wavestats/q PopenDif
            Residual[index2] = V_Avg //calculates the average residual per sampling point

            index2 += 1
            frqstr = num2str(freqwave[index2]) //these two commands update the index to check the next frequency
        while (index2 <=3) //stops after all desired frequencies have been checked (here, 4 frequencies)

        wavestats/q residual
        Checker[index3] = V_avg //stores the average residual for all frequencies

        index3 += 1
        rd=rawave[index3]*rc*re/(rb*rf) //now that index3 (and thus the current values for a) has been updated, this recalculates “d”

    while (index3 <= 21)  //stops after 21 vaules for a have been checked

    wavestats/q checker
    ra = rawave[v_Minloc] //finds the value for a that minimizes the residual
    store_a[index4] = (rawave[v_minloc]*5) //stores the value for a (in units of ms-1, converted back from 5 kHz sampling)
    rd = ra*rc*re/(rb*rf)




//next rate b, increasing/decreasing by 1%

    rbwave = {rb*.99,rb*.991,rb*.992,rb*.993,rb*.994,rb*.995,rb*.996,rb*.997,rb*.998,rb*.999,rb,rb*1.001,rb*1.002,rb*1.003,rb*1.004,rb*1.005,rb*1.006,rb*1.007,rb*1.008,rb*1.009,rb*1.01}
    checker = 0;  residual = 0;  index3 = 0
    rd=ra*rc*re/(rbwave[index3]*rf)

    do
        index2 = 0 ;   frqstr = num2str(freqwave[index2])
        Popen = 0; Pinact = 0; Pinact2 = 0; Pclosed = 0;  residual = 0

        do
            index = 0
            duplicate $("pressure"+frqstr) Pressure;  Pressure = Pressure*50
            Popen = 0; Pinact = 0; Pinact2 = 0;  Pclosed = 0;  Timewave[index] = 0
            Popen[index,index+delay]=0;   Pinact[index,index+delay]=0;   Pinact2[index,index+delay]=0
            Pclosed[index,index+delay]=1

            do
```

```
                index +=1
                Timewave[index] = index*.0002
                raPressure[index+delay]= ra*exp(pressure[index]/slope*-1)
                rePressure[index+delay] = re*exp(pressure[index]/slope)
                Popen[index+delay] = Popen[(index+delay-1)] + Pclosed[(index+delay-
1)]*(raPressure[index+delay]) + Pinact[(index+delay-1)]*rd - Popen[(index+delay-1)]*(rbwave[index3]+rc+rg)+
Pinact2[(index+delay-1)]*rh
                Pinact[index+delay] = PInact[(index+delay-1)] + Popen[(index+delay-1)]*rc +
Pclosed[(index+delay-1)]*rf - Pinact[(index+delay-1)]*(rd+rePressure[index+delay])
                Pinact2[index+delay] = Pinact2[index+delay-1] - Pinact2[(index+delay-1)]*rh +
Popen[(index+delay-1)]*rg
                Pclosed[index+delay] = Pclosed[(index+delay-1)] + PInact[(index+delay-
1)]*rePressure[index+delay] + Popen[(index+delay-1)]*rbwave[index3] - Pclosed[(index+delay-
1)]*(raPressure[index+delay]+rf)
            while (index <= 20000)

            killwaves Pressure
            index = 0
            Pressuretest[index] = -50
            Popentest = 0; Pinacttest = 0; Pclosedtest = 0; Popentest[index] = 0; Pinacttest[index] = 0
            Pinacttest2[index] = 0; Pclosedtest[index] = 1; Timewavetest[index] = 0

            do
                index +=1
                Timewavetest[index] = index*.00002
                Pressuretest[index] = -50
                raPtest[index]= ra*.1*exp(pressuretest[index]/slope*-1)
                rePtest[index] = re*.1*exp(pressuretest[index]/slope)
                Popentest[index] = Popentest[(index-1)] + Pclosedtest[(index-1)]*(raPtest[index]) +
Pinacttest[(index-1)]*rd*.1 - Popentest[(index-1)]*(rbwave[index3]*.1+rc*.1+rg*.1) + Pinacttest2[(index-1)]*rh*.1
                Pinacttest[index] = PInacttest[(index-1)] + Popentest[(index-1)]*rc*.1 + Pclosedtest[(index-
1)]*rf*.1 - Pinacttest[(index-1)]*(rd*.1+rePtest[index])
                Pinacttest2[index] = Pinacttest2[(index-1)] + Popentest[(index-1)]*rg*.1 - Pinacttest2[(index-
1)]*rh*.1
                Pclosedtest[index] = Pclosedtest[(index-1)] + PInacttest[(index-1)]*rePtest[index] +
Popentest[(index-1)]*rbwave[index3]*.1 - Pclosedtest[(index-1)]*(raPtest[index]+rf*.1)
            while (index <= 15000)

            wavestats/q Popentest
            scalefactor = V_max

            Duplicate/O $("avg_"+frqstr) realdata
            PopenDif = abs((Popen/scalefactor) - (realdata))
            killwaves realdata
            wavestats/q PopenDif
            Residual[index2] = V_Avg

            index2 += 1; frqstr = num2str(freqwave[index2])

        while (index2 <=3)

        wavestats/q residual
        Checker[index3] = V_avg

        index3 += 1
        rd=ra*rc*re/(rbwave[index3]*rf)
```

```
    while (index3 <= 21)  //stops after 21 vaules for a have been checked

    wavestats/q checker
    rb = rbwave[v_Minloc]; store_b[index4]= (rbwave[v_minloc]*5)
    rd=ra*rc*re/(rb*rf)


 //next rate c, increasing/decreasing by 1%
    rcwave =
{rc*.99,rc*.991,rc*.992,rc*.993,rc*.994,rc*.995,rc*.996,rc*.997,rc*.998,rc*.999,rc,rc*1.001,rc*1.002,rc*1.003,rc*
1.004,rc*1.005,rc*1.006,rc*1.007,rc*1.008,rc*1.009,rc*1.01}
    checker = 0; residual = 0; index3 = 0
    rd=ra*rcwave[index3]*re/(rb*rf)

    do
        index2 = 0; frqstr = num2str(freqwave[index2])
        Popen = 0; Pinact = 0; Pinact2 = 0;  Pclosed = 0;  residual = 0

        do
            index = 0
            duplicate $("pressure"+frqstr) Pressure;  Pressure = Pressure*50
            Popen = 0; Pinact = 0; Pinact2 = 0;  Pclosed = 0
            Timewave[index] = 0
            Popen[index,index+delay]=0;   Pinact[index,index+delay]=0; Pinact2[index,index+delay]=0
            Pclosed[index,index+delay]=1

            do
                index +=1
                Timewave[index] = index*.0002
                raPressure[index+delay]= ra*exp(pressure[index]/slope*-1)
                rePressure[index+delay] = re*exp(pressure[index]/slope)
                Popen[index+delay] = Popen[(index+delay-1)] + Pclosed[(index+delay-
1)]*(raPressure[index+delay]) + Pinact[(index+delay-1)]*rd - Popen[(index+delay-1)]*(rb+rcwave[index3]+rg) +
Pinact2[(index+delay-1)]*rh
                Pinact[index+delay] = PInact[(index+delay-1)] + Popen[(index+delay-1)]*rcwave[index3] +
Pclosed[(index+delay-1)]*rf - Pinact[(index+delay-1)]*(rd+rePressure[index+delay])
                Pinact2[index+delay] = Pinact2[(index+delay-1)] + Popen[(index+delay-1)]*rg -
Pinact2[(index+delay-1)]*rh
                Pclosed[index+delay] = Pclosed[(index+delay-1)] + PInact[(index+delay-
1)]*rePressure[index+delay] + Popen[(index+delay-1)]*rb - Pclosed[(index+delay-1)]*(raPressure[index+delay]+rf)
            while (index <= 20000)

            killwaves pressure
            index = 0
            Pressuretest[index] = -50
            Popentest = 0; Pinacttest = 0; Pinacttest2 = 0; Pclosedtest = 0;  Popentest[index] = 0
            Pinacttest[index] = 0;  Pinacttest2[index] = 0; Pclosedtest[index] = 1; Timewavetest[index] = 0

            do
                index +=1
                Timewavetest[index] = index*.00002
                Pressuretest[index] = -50
                raPtest[index]= ra*.1*exp(pressuretest[index]/slope*-1)
                rePtest[index] = re*.1*exp(pressuretest[index]/slope)
```

```
                Popentest[index] = Popentest[(index-1)] + Pclosedtest[(index-1)]*(raPtest[index]) +
Pinacttest[(index-1)]*rd*.1 - Popentest[(index-1)]*(rb*.1+rcwave[index3]*.1+rg*.1)+ Pinacttest2[(index-1)]*rh*.1
                Pinacttest[index] = PInacttest[(index-1)] + Popentest[(index-1)]*rcwave[index3]*.1+
Pclosedtest[(index-1)]*rf*.1 - Pinacttest[(index-1)]*(rd*.1+rePtest[index])
                Pinacttest2[index] = Pinacttest2[(index-1)] + Popentest[(index-1)]*rg*.1 - Pinacttest2[(index-
1)]*rh*.1
                Pclosedtest[index] = Pclosedtest[(index-1)] + PInacttest[(index-1)]*rePtest[index] +
Popentest[(index-1)]*rb*.1 - Pclosedtest[(index-1)]*(raPtest[index]+rf*.1)
            while (index <= 15000)

            wavestats/q Popentest
            scalefactor = V_max

            Duplicate/O $("avg_"+frqstr) realdata
            PopenDif = abs((Popen/scalefactor) - (realdata))
            killwaves realdata
            wavestats/q PopenDif
            Residual[index2] = V_Avg

            index2 += 1; frqstr = num2str(freqwave[index2])

        while (index2 <=3)
        wavestats/q residual
        Checker[index3] = V_avg

        index3 += 1
        rd=ra*rcwave[index3]*re/(rb*rf)

    while (index3 <= 21)

    wavestats/q checker
    rc = rcwave[v_Minloc]; store_c[index4]= (rcwave[v_minloc]*5)
    rd=ra*rc*re/(rb*rf)

 //next rate e , increasing/decreasing by 1%
    rewave =
{re*.99,re*.991,re*.992,re*.993,re*.994,re*.995,re*.996,re*.997,re*.998,re*.999,re,re*1.001,re*1.002,re*1.003,re*
1.004,re*1.005,re*1.006,re*1.007,re*1.008,re*1.009,re*1.01}
    checker = 0;  residual = 0;  index3 = 0
   rd=ra*rc*rewave[index3]/(rb*rf)

    do
        index2 = 0 ;   frqstr = num2str(freqwave[index2])
        Popen = 0; Pinact = 0; Pinact2 = 0; Pclosed = 0;  residual = 0

        do
            index = 0
            duplicate $("pressure"+frqstr) Pressure
            Pressure = Pressure*50
            Popen = 0; Pinact = 0; Pinact2 = 0; Pclosed = 0;  Timewave[index] = 0
            Popen[index,index+delay]=0; Pinact[index,index+delay]=0; Pinact2[index,index+delay]=0
            Pclosed[index,index+delay]=1

            do
                index +=1
                Timewave[index] = index*.0002
```

```
                raPressure[index+delay]= ra*exp(pressure[index]/slope*-1)
                rePressure[index+delay] = rewave[index3]*exp(pressure[index]/slope)
                Popen[index+delay] = Popen[(index+delay-1)] + Pclosed[(index+delay-
1)]*(raPressure[index+delay]) + Pinact[(index+delay-1)]*rd - Popen[(index+delay-1)]*(rb+rc+rg)+
Pinact2[(index+delay-1)]*rh
                Pinact[index+delay] = PInact[(index+delay-1)] + Popen[(index+delay-1)]*rc +
Pclosed[(index+delay-1)]*rf - Pinact[(index+delay-1)]*(rd+rePressure[index+delay])
                Pinact2[index+delay] = Pinact2[index+delay-1] + Popen[(index+delay-1)]*rg -
Pinact2[(index+delay-1)]*rh
                Pclosed[index+delay] = Pclosed[(index+delay-1)] + PInact[(index+delay-
1)]*rePressure[index+delay] + Popen[(index+delay-1)]*rb - Pclosed[(index+delay-1)]*(raPressure[index+delay]+rf)
            while (index <= 20000)

            killwaves pressure
            index = 0
            Pressuretest[index] = -50
            Popentest = 0; Pinacttest = 0; Pclosedtest = 0; Popentest[index] = 0; Pinacttest[index] = 0
            Pinacttest2[index] = 0; Pclosedtest[index] = 1; Timewavetest[index] = 0

            do
                index +=1
                Timewavetest[index] = index*.00002
                Pressuretest[index] = -50
                raPtest[index]= ra*.1*exp(pressuretest[index]/slope*-1)
                rePtest[index] = rewave[index3]*.1*exp(pressuretest[index]/slope)
                Popentest[index] = Popentest[(index-1)] + Pclosedtest[(index-1)]*(raPtest[index]) +
Pinacttest[(index-1)]*rd*.1- Popentest[(index-1)]*(rb*.1+rc*.1+rg*.1)+ Pinacttest2[(index-1)]*rh*.1
                Pinacttest[index] = PInacttest[(index-1)] + Popentest[(index-1)]*rc*.1 + Pclosedtest[(index-
1)]*rf*.1 - Pinacttest[(index-1)]*(rd*.1+rePtest[index])
                Pinacttest2[index] = Pinacttest2[(index-1)] + Popentest[(index-1)]*rg*.1- Pinacttest2[(index-
1)]*rh*.1
                Pclosedtest[index] = Pclosedtest[(index-1)] + PInacttest[(index-1)]*rePtest[index] +
Popentest[(index-1)]*rb*.1- Pclosedtest[(index-1)]*(raPtest[index]+rf*.1)
            while (index <= 15000)

            wavestats/q Popentest
            scalefactor = V_max

            Duplicate/O $("avg_"+frqstr) realdata
            PopenDif = abs((Popen/scalefactor) - (realdata))
            killwaves realdata
            wavestats/q PopenDif
            Residual[index2] = V_Avg

            index2 += 1; frqstr = num2str(freqwave[index2])

        while (index2 <=3)

        wavestats/q residual
        Checker[index3] = V_avg

        index3 += 1
        rd=ra*rc*rewave[index3]/(rb*rf)

    while (index3 <= 21)
    wavestats/q checker
```

```
    re = rewave[v_Minloc]; store_e[index4] = (rewave[v_minloc]*5)
    rd= ra*rc*re/(rb*rf)

//next rate f, increasing/decreasing by 1%

    rfwave =
{rf*.99,rf*.991,rf*.992,rf*.993,rf*.994,rf*.995,rf*.996,rf*.997,rf*.998,rf*.999,rf,rf*1.001,rf*1.002,rf*1.003,rf*1.00
4,rf*1.005,rf*1.006,rf*1.007,rf*1.008,rf*1.009,rf*1.01}
    checker = 0; residual = 0; index3 = 0
    rd=ra*rc*re/(rb*rfwave[index3])

    do
        index2 = 0; frqstr = num2str(freqwave[index2])
        Popen = 0; Pinact = 0; Pinact2 = 0;  Pclosed = 0;  residual = 0

        do
            index = 0
            duplicate $("pressure"+frqstr) Pressure
            Pressure = Pressure*50
            Popen = 0; Pinact = 0; Pinact2 = 0;  Pclosed = 0;  Timewave[index] = 0
            Popen[index,index+delay]=0;   Pinact[index,index+delay]=0; Pinact2[index,index+delay]=0
            Pclosed[index,index+delay]=1

            do
                index +=1
                Timewave[index] = index*.0002
                raPressure[index+delay]= ra*exp(pressure[index]/slope*-1)
                rePressure[index+delay] = re*exp(pressure[index]/slope)
                Popen[index+delay] = Popen[(index+delay-1)] + Pclosed[(index+delay-
1)]*(raPressure[index+delay]) + Pinact[(index+delay-1)]*rd - Popen[(index+delay-1)]*(rb+rc+rg)+
Pinact2[(index+delay-1)]*rh
                Pinact[index+delay] = PInact[(index+delay-1)] + Popen[(index+delay-1)]*rc +
Pclosed[(index+delay-1)]*rfwave[index3] - Pinact[(index+delay-1)]*(rd+rePressure[index+delay])
                Pinact2[index+delay] = Pinact2[index+delay-1] - Pinact2[(index+delay-1)]*rh +
Popen[(index+delay-1)]*rg
                Pclosed[index+delay] = Pclosed[(index+delay-1)] + PInact[(index+delay-
1)]*rePressure[index+delay] + Popen[(index+delay-1)]*rb - Pclosed[(index+delay-
1)]*(raPressure[index+delay]+rfwave[index3])
            while (index <= 20000)

            killwaves Pressure
            index = 0
            Pressuretest[index] = -50
            Popentest = 0; Pinacttest = 0; Pclosedtest = 0; Popentest[index] = 0
            Pinacttest[index] = 0;  Pinacttest2[index] = 0; Pclosedtest[index] = 1
            Timewavetest[index] = 0

            do
                index +=1
                Timewavetest[index] = index*.00002
                Pressuretest[index] = -50
                raPtest[index]= ra*.1*exp(pressuretest[index]/slope*-1)
                rePtest[index] = re*.1*exp(pressuretest[index]/slope)
                Popentest[index] = Popentest[(index-1)] + Pclosedtest[(index-1)]*(raPtest[index]) +
Pinacttest[(index-1)]*rd*.1 - Popentest[(index-1)]*(rb*.1+rc*.1+rg*.1)+ Pinacttest2[(index-1)]*rh*.1
```

```
                Pinacttest[index] = PInacttest[(index-1)] + Popentest[(index-1)]*rc*.1 + Pclosedtest[(index-1)]*rfwave[index3]*.1 - Pinacttest[(index-1)]*(rd*.1+rePtest[index])
                Pinacttest2[index] = Pinacttest2[(index-1)] + Popentest[(index-1)]*rg*.1 - Pinacttest2[(index-1)]*rh*.1
                Pclosedtest[index] = Pclosedtest[(index-1)] + PInacttest[(index-1)]*rePtest[index] + Popentest[(index-1)]*rb*.1 - Pclosedtest[(index-1)]*(raPtest[index]+rfwave[index3]*.1)
            while (index <= 15000)

            wavestats/q Popentest
            scalefactor = V_max

            Duplicate/O $("avg_"+frqstr) realdata
            PopenDif = abs((Popen/scalefactor) - (realdata))
            killwaves realdata
            wavestats/q PopenDif
            Residual[index2] = V_Avg

            index2 += 1; frqstr = num2str(freqwave[index2])

        while (index2 <=3) //stops after all 7 frequencies have been checked

        wavestats/q residual
        Checker[index3] = V_avg
        index3 += 1
        rd=ra*rc*re/(rb*rfwave[index3])

    while (index3 <= 21)

    wavestats/q checker
    rf = rfwave[v_Minloc]; store_f[index4]= (rfwave[v_minloc]*5)
    rd=ra*rc*re/(rb*rf)

    //next rate g, increasing/decreasing by 1%
    rgwave = {rg*.99,rg*.991,rg*.992,rg*.993,rg*.994,rg*.995,rg*.996,rg*.997,rg*.998,rg*.999,rg,rg*1.001,rg*1.002,rg*1.003,rg*1.004,rg*1.005,rg*1.006,rg*1.007,rg*1.008,rg*1.009,rg*1.01}
    checker = 0; residual = 0; index3 = 0
    rd=ra*rc*re/(rf*rb)

    do
        index2 = 0;    frqstr = num2str(freqwave[index2])
        Popen = 0; Pinact = 0; Pinact2 = 0; Pclosed = 0;  residual = 0

        do
            index = 0
            duplicate $("pressure"+frqstr) Pressure
            Pressure = Pressure*50
            Popen = 0; Pinact = 0; Pinact2 = 0; Pclosed = 0;  Popen[index] = 0
            Pinact[index] = 0; Pclosed[index] = 1;Timewave[index] = 0
            Popen[index,index+delay]=0;   Pinact[index,index+delay]=0;   Pinact2[index,index+delay]=0
            Pclosed[index,index+delay]=1

            do
                index +=1
                Timewave[index] = index*.0002
                raPressure[index+delay]= ra*exp(pressure[index]/slope*-1)
```

```
                rePressure[index+delay] = re*exp(pressure[index]/slope)
                Popen[index+delay] = Popen[(index+delay-1)] + Pclosed[(index+delay-
1)]*(raPressure[index+delay]) + Pinact[(index+delay-1)]*rd - Popen[(index+delay-1)]*(rb+rc+rgwave[index3])+
Pinact2[(index+delay-1)]*rh
                Pinact[index+delay] = PInact[(index+delay-1)] + Popen[(index+delay-1)]*rc +
Pclosed[(index+delay-1)]*rf - Pinact[(index+delay-1)]*(rd+rePressure[index+delay])
                Pinact2[index+delay] = Pinact2[(index+delay-1)] + Popen[(index+delay-1)]*rgwave[index3] -
Pinact2[(index+delay-1)]*rh
                Pclosed[index+delay] = Pclosed[(index+delay-1)] + PInact[(index+delay-
1)]*rePressure[index+delay] + Popen[(index+delay-1)]*rb - Pclosed[(index+delay-1)]*(raPressure[index+delay]+rf)
            while (index <= 20000)

            killwaves pressure
            index = 0
            Pressuretest[index] = -50
            Popentest = 0; Pinacttest = 0; Pinacttest2 = 0; Pclosedtest = 0; Popentest[index] = 0;
            Pinacttest[index] = 0;  Pinacttest2[index] = 0; Pclosedtest[index] = 1; Timewavetest[index] = 0

            do
                index +=1
                Timewavetest[index] = index*.00002
                Pressuretest[index] = -50
                raPtest[index]= ra*.1*exp(pressuretest[index]/slope*-1)
                rePtest[index] = re*.1*exp(pressuretest[index]/slope)
                Popentest[index] = Popentest[(index-1)] + Pclosedtest[(index-1)]*(raPtest[index]) +
Pinacttest[(index-1)]*rd*.1 - Popentest[(index-1)]*(rb*.1+rc*.1+rgwave[index3]*.1)+ Pinacttest2[(index-1)]*rh*.1
                Pinacttest[index] = PInacttest[(index-1)] + Popentest[(index-1)]*rc*.1 + Pclosedtest[(index-
1)]*rf*.1 - Pinacttest[(index-1)]*(rd*.1+rePtest[index])
                Pinacttest2[index] = Pinacttest2[index-1] + Popentest[(index-1)]*rgwave[index3]*.1-
Pinacttest2[(index-1)]*rh*.1
                Pclosedtest[index] = Pclosedtest[(index-1)] + PInacttest[(index-1)]*rePtest[index] +
Popentest[(index-1)]*rb*.1- Pclosedtest[(index-1)]*(raPtest[index]+rf*.1)
            while (index <= 15000)

            wavestats/q Popentest
            scalefactor = V_max
            Duplicate/O $("avg_"+frqstr) realdata
            PopenDif = abs((Popen/scalefactor) - (realdata))
            killwaves realdata
            wavestats/q PopenDif
            Residual[index2] = V_Avg

            index2 += 1; frqstr = num2str(freqwave[index2])

        while (index2 <=3) //stops after all 7 frequencies have been checked

        wavestats/q residual
        Checker[index3] = V_avg
        index3 += 1
        rd=ra*rc*re/(rb*rf)

    while (index3 <= 21)

    wavestats/q checker
    rg = rgwave[v_Minloc]; store_g[index4] = (rgwave[v_minloc]*5)
    rd=ra*rc*re/(rb*rf)
```

```
    //next rate h, increasing/decreasing by 1%
    rhwave =
{rh*.99,rh*.991,rh*.992,rh*.993,rh*.994,rh*.995,rh*.996,rh*.997,rh*.998,rh*.999,rh,rh*1.001,rh*1.002,rh*1.003,rh
*1.004,rh*1.005,rh*1.006,rh*1.007,rh*1.008,rh*1.009,rh*1.01}
    checker = 0; residual = 0; index3 = 0
    rd=ra*rc*re/(rb*rf)

    do
        index2 = 0 ;  frqstr = num2str(freqwave[index2])
        Popen = 0; Pinact = 0; Pinact2 = 0; Pclosed = 0; residual = 0

        do
            index = 0
            duplicate $("pressure"+frqstr) Pressure; Pressure = Pressure*50
            Popen = 0; Pinact = 0; Pinact2 = 0;  Pclosed = 0;  Popen[index] = 0; Pinact[index] = 0
            Pclosed[index] = 1; Timewave[index] = 0; Popen[index,index+delay]=0;
            Pinact[index,index+delay]=0; Pinact2[index,index+delay]=0;  Pclosed[index,index+delay]=1

            do
                index +=1
                Timewave[index] = index*.0002
                raPressure[index+delay]= ra*exp(pressure[index]/slope*-1)
                rePressure[index+delay] = re*exp(pressure[index]/slope)
                Popen[index+delay] = Popen[(index+delay-1)] + Pclosed[(index+delay-
1)]*(raPressure[index+delay]) + Pinact[(index+delay-1)]*rd - Popen[(index+delay-1)]*(rb+rc+rg)+
Pinact2[(index+delay-1)]*rhwave[index3]
                Pinact[index+delay] = PInact[(index+delay-1)] + Popen[(index+delay-1)]*rc +
Pclosed[(index+delay-1)]*rf - Pinact[(index+delay-1)]*(rd+rePressure[index+delay])
                Pinact2[index+delay] = Pinact2[(index+delay-1)] + Popen[(index+delay-1)]*rg -
Pinact2[(index+delay-1)]*rhwave[index3]
                Pclosed[index+delay] = Pclosed[(index+delay-1)] + PInact[(index+delay-
1)]*rePressure[index+delay] + Popen[(index+delay-1)]*rb - Pclosed[(index+delay-1)]*(raPressure[index+delay]+rf)
            while (index <= 20000)

            killwaves pressure
            index = 0
            Pressuretest[index] = -50
            Popentest = 0; Pinacttest = 0; Pinacttest2 = 0; Pclosedtest = 0; Popentest[index] = 0
            Pinacttest[index] = 0;  Pinacttest2[index] = 0; Pclosedtest[index] = 1; Timewavetest[index] = 0

            do
                index +=1
                Timewavetest[index] = index*.00002
                Pressuretest[index] = -50
                raPtest[index]= ra*.1*exp(pressuretest[index]/slope*-1)
                rePtest[index] = re*.1*exp(pressuretest[index]/slope)
                Popentest[index] = Popentest[(index-1)] + Pclosedtest[(index-1)]*(raPtest[index]) +
Pinacttest[(index-1)]*rd*.1 - Popentest[(index-1)]*(rb*.1+rc*.1+rg*.1)+ Pinacttest2[(index-1)]*rhwave[index3]*.1
                Pinacttest[index] = PInacttest[(index-1)] + Popentest[(index-1)]*rc*.1 + Pclosedtest[(index-
1)]*rf*.1 - Pinacttest[(index-1)]*(rd*.1+rePtest[index])
                Pinacttest2[index] = Pinacttest2[index-1] + Popentest[(index-1)]*rg*.1 - Pinacttest2[(index-
1)]*rhwave[index3]*.1
                Pclosedtest[index] = Pclosedtest[(index-1)] + PInacttest[(index-1)]*rePtest[index] +
Popentest[(index-1)]*rb*.1 - Pclosedtest[(index-1)]*(raPtest[index]+rf*.1)
            while (index <= 15000)
```

```
            wavestats/q Popentest
            scalefactor = V_max
            Duplicate/O $("avg_"+frqstr) realdata
            PopenDif = abs((Popen/scalefactor) - (realdata))
            killwaves realdata
            wavestats/q PopenDif
            Residual[index2] = V_Avg

            index2 += 1; frqstr = num2str(freqwave[index2])

        while (index2 <=3)

        wavestats/q residual
        Checker[index3] = V_avg
        index3 += 1
        rd=ra*rc*re/(rb*rf)

    while (index3 <= 21)

    wavestats/q checker
    rh = rhwave[v_Minloc]; store_h[index4] = (rhwave[v_minloc]*5)
    rd=ra*rc*re/(rb*rf)


 //next rate slope, increasing/decreasing by 1%
    slopewave =
{slope*.99,slope*.991,slope*.992,slope*.993,slope*.994,slope*.995,slope*.996,slope*.997,slope*.998,slope*.999,sl
ope,slope*1.001,slope*1.002,slope*1.003,slope*1.004,slope*1.005,slope*1.006,slope*1.007,slope*1.008,slope*1.0
09,slope*1.01}
    checker = 0; residual = 0; index3 = 0

    do
        index2 = 0; frqstr = num2str(freqwave[index2])
        Popen = 0; Pinact = 0; Pinact2 = 0;  Pclosed = 0;  residual = 0

        do
            index = 0
            duplicate $("pressure"+frqstr) Pressure;  Pressure = Pressure*50
            Popen = 0; Pinact = 0; Pinact2 = 0;  Pclosed = 0;Timewave[index] = 0
            Popen[index,index+delay]=0; Pinact[index,index+delay]=0; Pinact2[index,index+delay]=0
            Pclosed[index,index+delay]=1

            do
                index +=1
                Timewave[index] = index*.0002
                raPressure[index+delay]= ra*exp(pressure[index]/slopewave[index3]*-1)
                rePressure[index+delay] = re*exp(pressure[index]/slopewave[index3])
                Popen[index+delay] = Popen[(index+delay-1)] + Pclosed[(index+delay-
1)]*(raPressure[index+delay]) + Pinact[(index+delay-1)]*rd - Popen[(index+delay-1)]*(rb+rc+rg)+
Pinact2[(index+delay-1)]*rh
                Pinact[index+delay] = PInact[(index+delay-1)] + Popen[(index+delay-1)]*rc +
Pclosed[(index+delay-1)]*rf- Pinact[(index+delay-1)]*(rd+rePressure[index+delay])
                Pinact2[(index+delay)] = Pinact2[(index+delay-1)] + Popen[(index+delay-1)]*rg -
Pinact2[(index+delay-1)]*rh
```

```
                Pclosed[index+delay] = Pclosed[(index+delay-1)] + PInact[(index+delay-
1)]*rePressure[index+delay] + Popen[(index+delay-1)]*rb - Pclosed[(index+delay-1)]*(raPressure[index+delay]+rf)
            while (index <= 20000)

            killwaves pressure
            index = 0
            Pressuretest[index] = -50
            Popentest = 0; Pinacttest = 0; Pinacttest2 = 0; Pclosedtest = 0;  Popentest[index] = 0
            Pinacttest[index] = 0;  Pinacttest2[index] = 0; Pclosedtest[index] = 1; Timewavetest[index] = 0

            do
                index +=1
                Timewavetest[index] = index*.00002
                Pressuretest[index] = -50
                raPtest[index]= ra*.1*exp(pressuretest[index]/slopewave[index3]*-1)
                rePtest[index] = re*.1*exp(pressuretest[index]/slopewave[index3])
                Popentest[index] = Popentest[(index-1)] + Pclosedtest[(index-1)]*(raPtest[index]) +
Pinacttest[(index-1)]*rd*.1 - Popentest[(index-1)]*(rb*.1+rc*.1+rg*.1)+ Pinacttest2[(index-1)]*rh*.1
                Pinacttest[index] = PInacttest[(index-1)] + Popentest[(index-1)]*rc*.1 + Pclosedtest[(index-
1)]*rf*.1 - Pinacttest[(index-1)]*(rd*.1+rePtest[index])
                Pinacttest2[index] = Pinacttest2[(index-1)] + Popentest[(index-1)]*rg*.1- Pinacttest2[(index-
1)]*rh*.1
                Pclosedtest[index] = Pclosedtest[(index-1)] + PInacttest[(index-1)]*rePtest[index] +
Popentest[(index-1)]*rb*.1 - Pclosedtest[(index-1)]*(raPtest[index]+rf*.1)
            while (index <= 15000)

            wavestats/q Popentest
            scalefactor = V_max
            Duplicate/O $("avg_"+frqstr) realdata
            PopenDif = abs((Popen/scalefactor) - (realdata))
            killwaves realdata
            wavestats/q PopenDif
            Residual[index2] = V_Avg

            index2 += 1; frqstr = num2str(freqwave[index2])

        while (index2 <=3)

        wavestats/q residual
        Checker[index3] = V_avg
        index3 += 1

    while (index3 <= 21)  //stops after 21 vaules for a have been checked

    wavestats/q checker
    slope= slopewave[v_Minloc];   store_slope[index4] = slopewave[v_minloc]
    rd=ra*rc*re/(rb*rf)

 //next rate delay, varying from 0 to 20 points
    delaywave = {0,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40}
    checker = 0; residual = 0; index3 = 0

    do
        index2 = 0; frqstr = num2str(freqwave[index2])
        Popen = 0; Pinact = 0; Pinact2 = 0;  Pclosed = 0; residual = 0
```

```
	do
		index = 0
		duplicate $("pressure"+frqstr) Pressure; Pressure = Pressure*50
		Popen = 0; Pinact = 0; Pinact2 = 0; Pclosed = 0; Timewave[index] = 0
		Popen[index,index+delaywave[index3]]=0;  Pinact[index,index+delaywave[index3]]=0
		Pinact2[index,index+delaywave[index3]]=0; Pclosed[index,index+delaywave[index3]]=1

		do
			index +=1
			Timewave[index] = index*.0002
			raPressure[index+delaywave[index3]]= ra*exp(pressure[index]/slope*-1)
			rePressure[index+delaywave[index3]] = re*exp(pressure[index]/slope)
			Popen[index+delaywave[index3]] = Popen[(index+delaywave[index3]-1)] +
Pclosed[(index+delaywave[index3]-1)]*(raPressure[index+delaywave[index3]]) +
Pinact[(index+delaywave[index3]-1)]*rd - Popen[(index+delaywave[index3]-1)]*(rb+rc+rg)+
Pinact2[(index+delaywave[index3]-1)]*rh
			Pinact[index+delaywave[index3]] = PInact[(index+delaywave[index3]-1)] +
Popen[(index+delaywave[index3]-1)]*rc + Pclosed[(index+delaywave[index3]-1)]*rf-
Pinact[(index+delaywave[index3]-1)]*(rd+rePressure[index+delaywave[index3]])
			Pinact2[index+delaywave[index3]] = Pinact2[index+delaywave[index3]-1] +
Popen[(index+delaywave[index3]-1)]*rg - Pinact2[(index+delaywave[index3]-1)]*rh
			Pclosed[index+delaywave[index3]] = Pclosed[(index+delaywave[index3]-1)] +
PInact[(index+delaywave[index3]-1)]*rePressure[index+delaywave[index3]] + Popen[(index+delaywave[index3]-
1)]*rb - Pclosed[(index+delaywave[index3]-1)]*(raPressure[index+delaywave[index3]]+rf)
		while (index <= 20000)

		killwaves pressure
		index = 0
		Pressuretest[index] = -50
		Popentest = 0; Pinacttest = 0; Pinacttest2 = 0; Pclosedtest = 0; Popentest[index] = 0
		Pinacttest[index] = 0; Pinacttest2[index] = 0; Pclosedtest[index] = 1; Timewavetest[index] = 0

		do
			index +=1
			Timewavetest[index] = index*.00002
			Pressuretest[index] = -50
			raPtest[index]= ra*.1*exp(pressuretest[index]/slope*-1)
			rePtest[index] = re*.1*exp(pressuretest[index]/slope)
			Popentest[index] = Popentest[(index-1)] + Pclosedtest[(index-1)]*(raPtest[index]) +
Pinacttest[(index-1)]*rd*.1- Popentest[(index-1)]*(rb*.1+rc*.1+rg*.1)+ Pinacttest2[(index-1)]*rh*.1
			Pinacttest[index] = PInacttest[(index-1)] + Popentest[(index-1)]*rc*.1 + Pclosedtest[(index-
1)]*rf*.1 - Pinacttest[(index-1)]*(rd*.1+rePtest[index])
			Pinacttest2[index] = Pinacttest2[(index-1)] + Popentest[(index-1)]*rg*.1 - Pinacttest2[(index-
1)]*rh*.1
			Pclosedtest[index] = Pclosedtest[(index-1)] + PInacttest[(index-1)]*rePtest[index] +
Popentest[(index-1)]*rb*.1 - Pclosedtest[(index-1)]*(raPtest[index]+rf*.1)
		while (index <= 15000)

		wavestats/q Popentest
		scalefactor = V_max
		Duplicate/O $("avg_"+frqstr) realdata
		PopenDif = abs((Popen/scalefactor) - (realdata))
		killwaves realdata
		wavestats/q PopenDif
		Residual[index2] = V_Avg
		Duplicate/O Popen $("Popen_"+frqstr)
```

```
            index2 += 1; frqstr = num2str(freqwave[index2])

        while (index2 <=3)

        wavestats/q residual
        Checker[index3] = V_avg

        index3 += 1

    while (index3 <= 21)

    wavestats/q checker
    delay= delaywave[v_Minloc]
    store_delay[index4] = delaywave[v_minloc]
    rd=ra*rc*re/(rb*rf)

  //this part needs to go after everything; it will store the values for d as well as print a few key results so that one can keep track of how the fit is progressing
     print index4 //prints iteration number
     print "residual", V_Min //prints the current residual; if it stops getting smaller, one can abort the procedure
    print "delay", store_delay[index4]
    store_d[index4]=rd*5;  print "rated" ,store_d[index4]
    store_residual[index4]=v_min
    index4 += 1

while (index4 <= 1000)

end
```