# Performing IV calculations within CMDF

Christopher George, Markus Buehler, and William Goddard

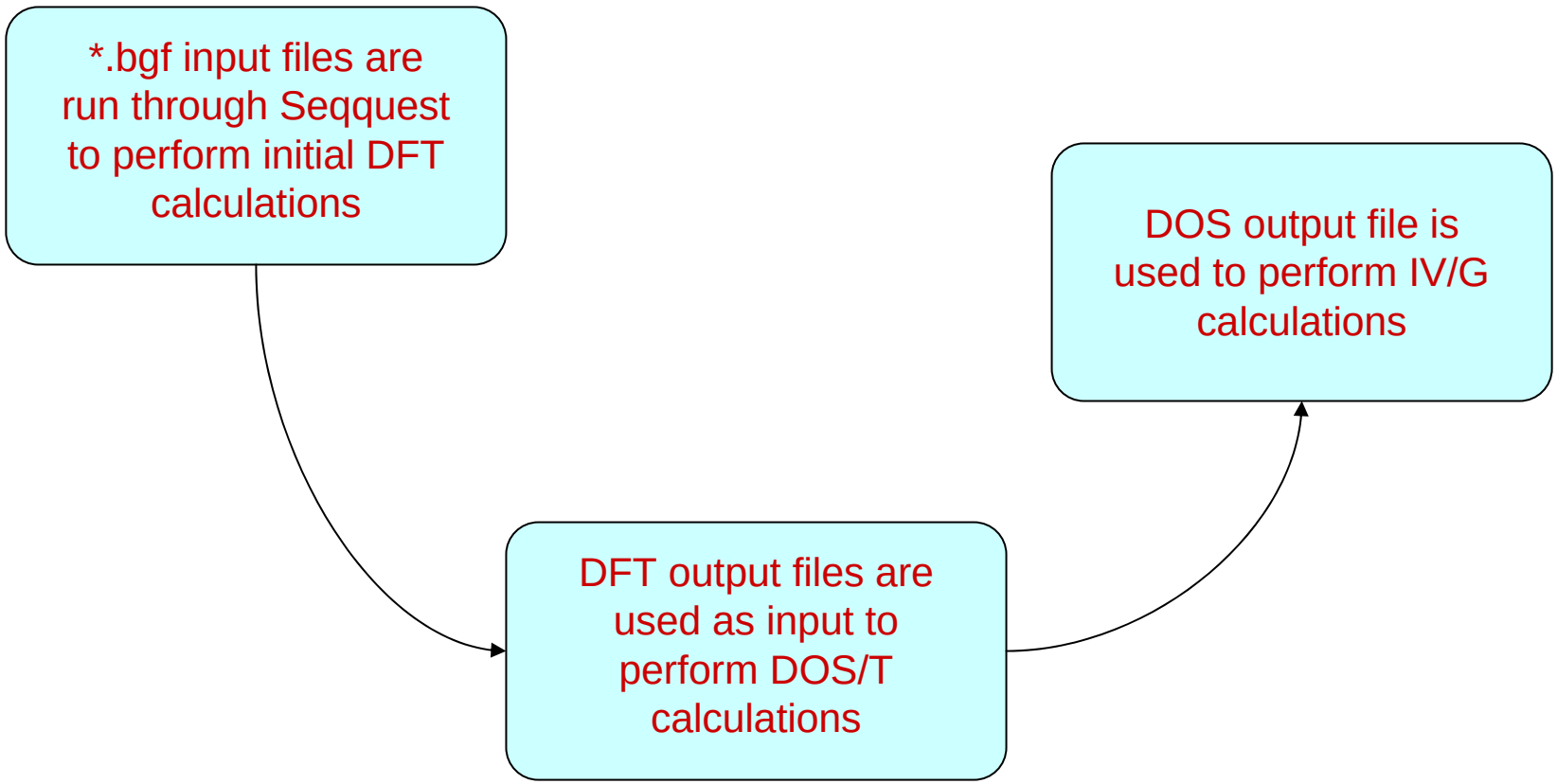
Materials and Process Simulation Center, California Institute of Technology

August 23, 2005

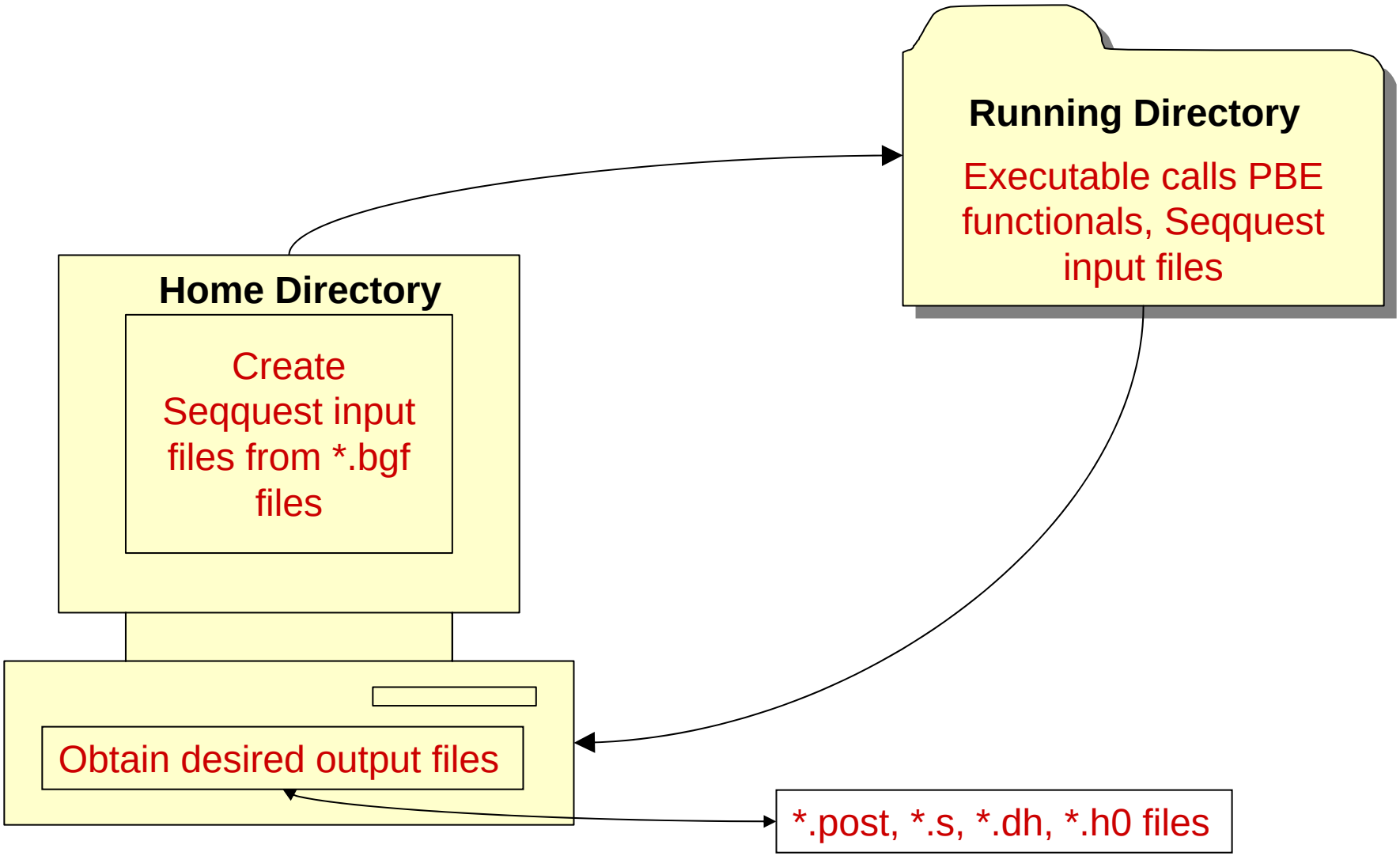
# Contents

- Yong-Hoon Kim's IV code
  - DFT
    - Input/Output
  - DOS calculation
  - IV calculation
- IV code in CMDF
  - Input/Output files
- Results

# Original IV Code

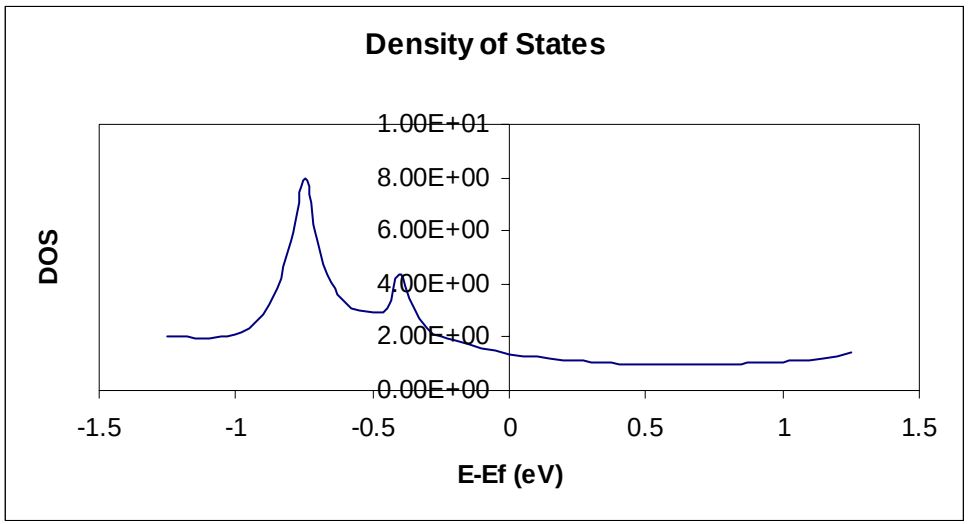


# DFT Calculations



# DOS/T Calculations

```
#  S [Ang^2]     Ef [eV]     Eg [eV]    Ef1 [eV]    Eg1 [eV]    Ef2 [eV]    Eg2 [eV]
#     43.215    -4.72243     0.01686    -4.41267     0.02750    -4.41224     0.02700
#  E-Ef [eV]      E [eV]           T       DOS [1/eV]          c_1
    -1.50000    -6.22243  1.4614E+00  7.4180E+00  3.7551E-02
    -1.49000    -6.21243  1.4660E+00  7.3887E+00  3.7352E-02
    -1.48000    -6.20243  1.4693E+00  7.3658E+00  3.7189E-02
    -1.47000    -6.19243  1.4712E+00  7.3463E+00  3.7050E-02
```

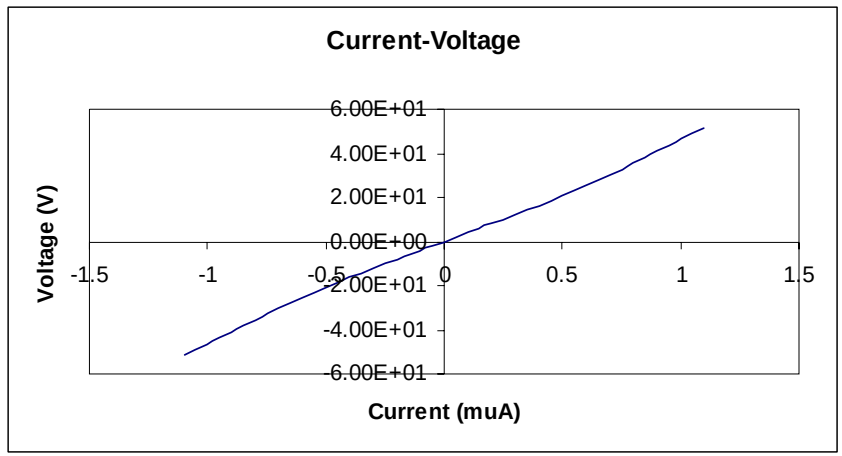



- DOS input file is created
  - DOS projection on atoms/orbitals
  - Minimum and maximum energy
  - Additional parameters
- Second executable sums *.dh and *.h0 files to create *.h files
- *.post, *.s, *.h files are copied to running directory to perform DOS/T calculations

# IV/GV Calculations

- IV input file is created
  - Minimum and maximum voltage
  - Voltage step size
  - Voltage distribution over electrodes
- Executable is run that takes in DOS output and IV input file
- Final output has information on current's and conductivity's dependence on voltage

```
# 2e^2/h (G0) =    77.48092 [muA/V] ; 2e/h =  2108.36294 [muA]
# vrat1=     0.50000
#      V [V]       I [muA]     G [muA/V]        I [G0]        G [G0]
      -1.250    -7.724E+01     6.477E+01    -9.968E-01     8.360E-01
      -1.245    -7.691E+01     6.472E+01    -9.927E-01     8.353E-01
      -1.240    -7.659E+01     6.468E+01    -9.885E-01     8.347E-01
      -1.235    -7.627E+01     6.461E+01    -9.843E-01     8.339E-01
```



# Implementation in CMDF

- Want to simplify process
  - Eliminate manually creating input files and running numerous executables
- Keep flexibility of code
  - Try not to lose any functionality
- Take advantage of CMDF
  - Framework creates a unique environment in which disorder from temperature effects are easily studied—make use of this

# Input for Python Code

- Three executables to run
  - Seqquestin.py
    - Creates Seqquest input
  - Projectionrunner.py
    - Gives atomic coordinates in Bohrs
    - Used to create projectioninput file
  - ManualIV.py
    - Main program that actually performs calculations

```
title = "testrun"
kgridwhole = "2 4 0"
kgridelect = "2 4 3"
convergence = "0.0001"
history = "10"
gsteps = "100"
gconv = "0.0005"
```

```
title = 'testrun'                    ## same as title used for seqquestin.py file
outputfile = 'ivout.txt'             ## file where output will be given
tempfold = "/temp1/ch121chg"         ## running folder for calculations
totalnumber = 56                     ## total number of atoms in the system
elect1number = 18                    ## number of atoms in the bottom electrode
elect2number = 18                    ## number of atoms in the top electrode
energymin = "-1.50"                  ## minimum energy for the Density of States
energymax = "1.50"                   ## maximum energy for the Density of States
energystep = "0.50"                  ## step size for the Density of States Calculation
numberenergylevels = "1"             ## number of energy levels per SGF block
broadening = "0.10"                  ## energy broadening (eV) for electrode
minvolt = "-1.50"                    ## minimum voltage to be calculated
maxvolt = "1.50"                     ## maximum voltage to be calculated
voltstep = "0.50"                    ## voltage step size
contactdist = "0.5"                  ## voltage distribution over contacts
```

# Python Code Output Files

## Seqquestin.py

```
do setup
do iters
do post
no force
no relax
no cell
no neb
setup data
title
 ...
functional
 PBE
dimension
3
(primitive lattice vectors …)
(grid dimensions…)
(atom types …)
(atom files …)
number of atoms in unit cell
 56
(atom, type, position vector …)
kgridh
 2 4 0
end setup phase data
run phase data
convergence
 0.0001
history
 10
geometry parameters
gsteps
 100
gconv
 0.0005
end geometry parameters
end run phase data
```

## Projectionrunner.py allows for the creation of:

```
number of projection
 2
projection name...1:
 c_1
no orbital
number of selected atoms
 8
selected atoms number
 10   2     0.088439218     0.403721250     2.844643664
 11   2     2.754258027     0.450208531     2.844643664
 12   2     0.007842367     5.021042109     2.844643664
 13   2     2.673661176     5.067548288     2.844643664
 14   2     4.127465858    -1.835094874     2.844643664
 15   2     6.793284667    -1.788588695     2.844643664
 16   2     4.046869007     2.782131499     2.844643664
 17   2     6.712687816     2.828637677     2.844643664
projection name...2:
 pt_bottom_p
do orbital
number of selected atoms
 9
selected atoms number
 1  0  1 1 1  0 0 0 0 0
 2  0  1 1 1  0 0 0 0 0
 3  0  1 1 1  0 0 0 0 0
 4  0  1 1 1  0 0 0 0 0
 5  0  1 1 1  0 0 0 0 0
 6  0  1 1 1  0 0 0 0 0
 7  0  1 1 1  0 0 0 0 0
 8  0  1 1 1  0 0 0 0 0
 9  0  1 1 1  0 0 0 0 0
```

## ManualIV.py

IV output for testrun

Density of States data

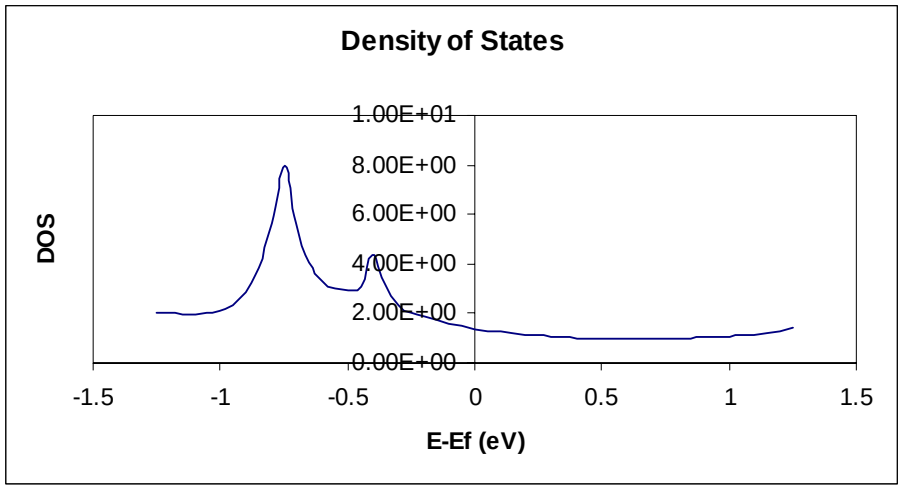
```
# dosrunner
#  S [Ang^2]    Ef [eV]    Eg [eV]    Ef1 [eV]    Eg1 [eV]    Ef2 [eV]
Eg2 [eV]
#     43.215    -4.83539     0.01565    -4.41227     0.02704    -
4.41230     0.02714
#  E-Ef [eV]      E [eV]          T  DOS [1/eV] c_1
   -1.25000    -6.08539  6.0383E-01  2.0322E+00  1.1249E-01
   -1.20000    -6.03539  6.0901E-01  1.9929E+00  1.1819E-01
   -1.15000    -5.98539  6.2323E-01  1.9754E+00  1.2563E-01
   -1.10000    -5.93539  6.4680E-01  1.9850E+00  1.3530E-01
   …
    1.05000    -3.78539  3.2641E-01  1.1163E+00  8.3698E-02
    1.10000    -3.73539  3.3791E-01  1.1566E+00  8.8980E-02
    1.15000    -3.68539  3.5250E-01  1.2202E+00  9.6814E-02
    1.20000    -3.63539  3.7272E-01  1.3136E+00  1.0815E-01
    1.25000    -3.58539  4.0116E-01  1.4467E+00  1.2427E-01
```

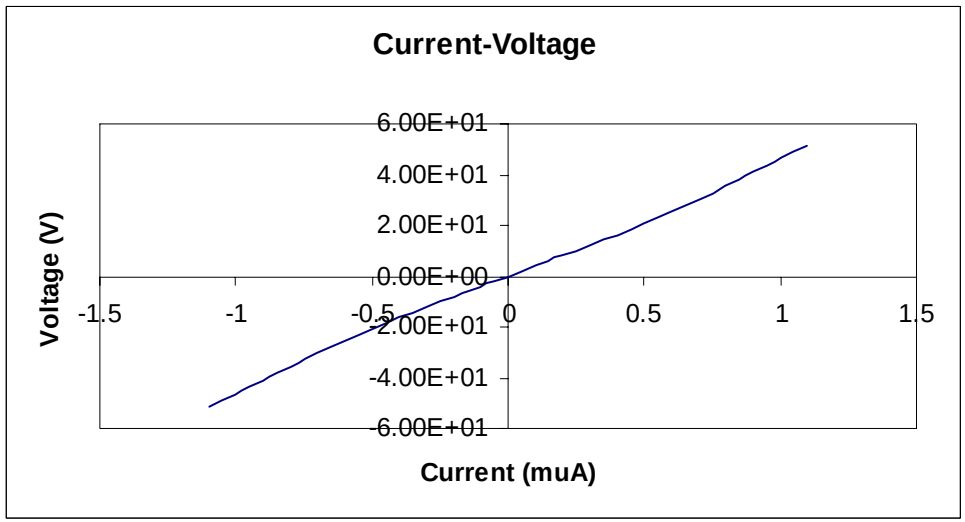
Current-Voltage data

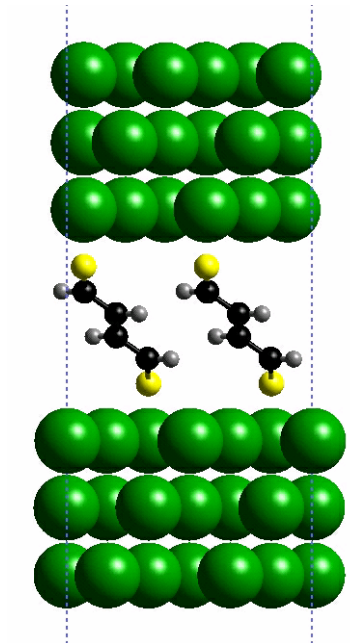
```
# 2e^2/h (G0) =    77.48092 [muA/V] ; 2e/h =  2108.36294 [muA]
# vrat1=     0.50000
#      V [V]       I [muA]     G [muA/V]        I [G0]        G [G0]
     -1.100    -5.165E+01     4.892E+01    -6.666E-01     6.313E-01
     -1.050    -4.897E+01     5.419E+01    -6.320E-01     6.994E-01
     -1.000    -4.629E+01     5.338E+01    -5.974E-01     6.890E-01
     …
      1.000     4.629E+01     5.338E+01     5.974E-01     6.890E-01
      1.050     4.897E+01     5.419E+01     6.320E-01     6.994E-01
      1.100     5.165E+01     4.892E+01     6.666E-01     6.313E-01
```

# Results

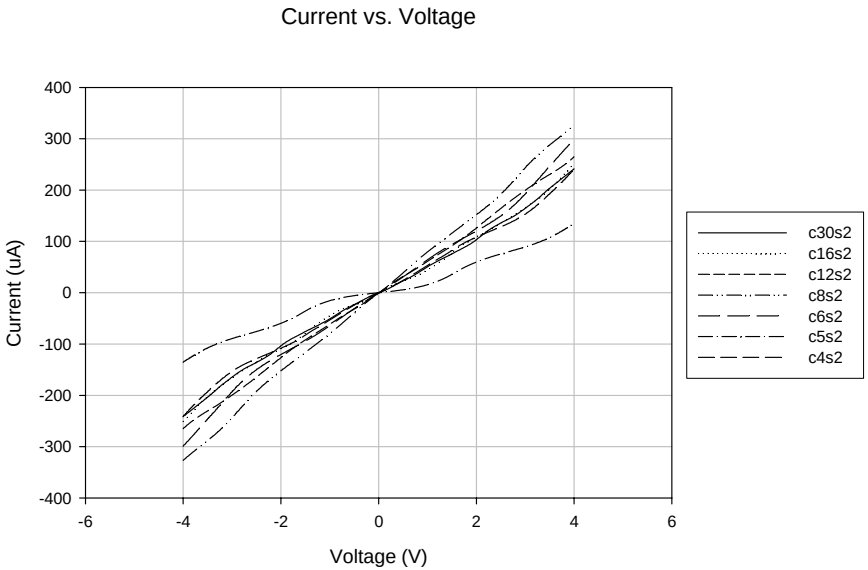
Results for $C_4H_4S_2$ between Au electrodes



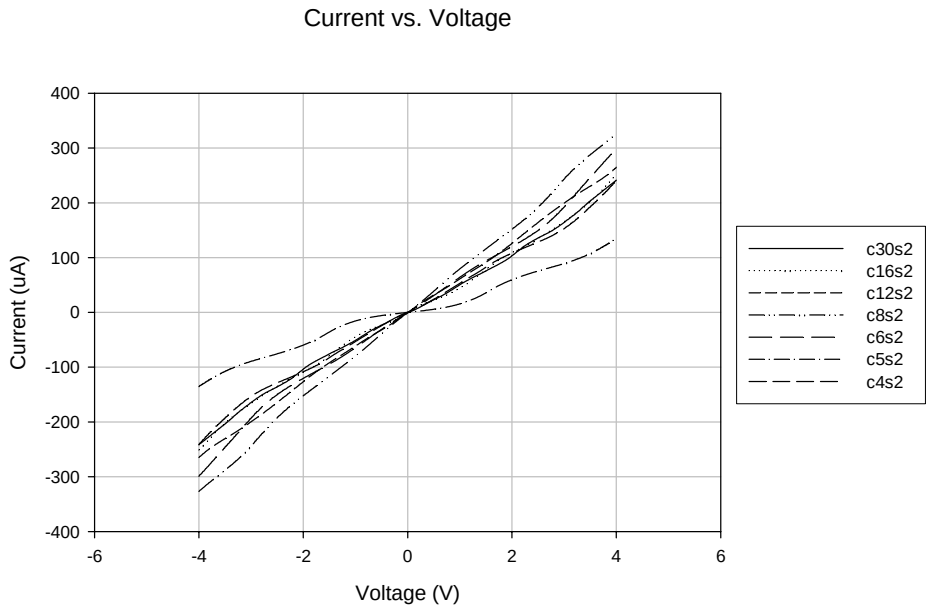



# Results (cont.)




**Alkane Dithiol Chains**

**Alkene Dithiol Chains**



# Upcoming Research

- Take advantage of the ease of creating systems that have undergone MD simulations
  - IV code requires periodic boundary conditions
  - REAXFF PBC's exist for CMDF, but no parameters for alkane/alkene systems
  - DREIDING PBC's are being developed
- Look at temperature effects of doped and undoped monolayers using CMDF MD simulations

# Summary

- Created Python wrapper for IV code
  - DFT calculations
  - DOS calculations
  - IV calculations
- Integrated into CMDF to allow for easy transition between MD simulations and IV calculations
- Looking to make use of this relationship in further research

# Acknowledgments

PROM, ASC, and NSF-ITR

The facilities of the Materials and Process Simulation Center are also supported by
DOE (ASCI ASAP),
NSF (CHE),
MURI-ARO,
MURI-ONR,
ChevronTexaco,
Seiko-Epson,
Beckman Institute,
and Asahi Kasei.