# VCSMD

This is just a little guide for VCSMD code. The code is written in Fortran 77 and you can find a reference here.

To compile the `celq.f` code, you need to run

```
gfortran -o <exec-name> celq.f
```

Then you `grep` one input from `AllInput.txt`, name it `inp` (it is important!), andrun them with

```
<path-to-your-exec>/<exec-name> inp
```

Make sure you have the file `sip` in the same folder at where your executable locates, or you will not be able to run. This file is the pair potential of $\mathrm{Si}$, if you are running $\mathrm{Ar}$, you do not need to use it. So you can comment the $75$th line in `celq.f`, i.e.,

```
open(unit=31,file='sip',status='old',form='formatted')
```

And for line $360$, if you want to get a clear `e` file as output, you'd better modify this line from

```
101   format(1x,4d12.5,i6)
```

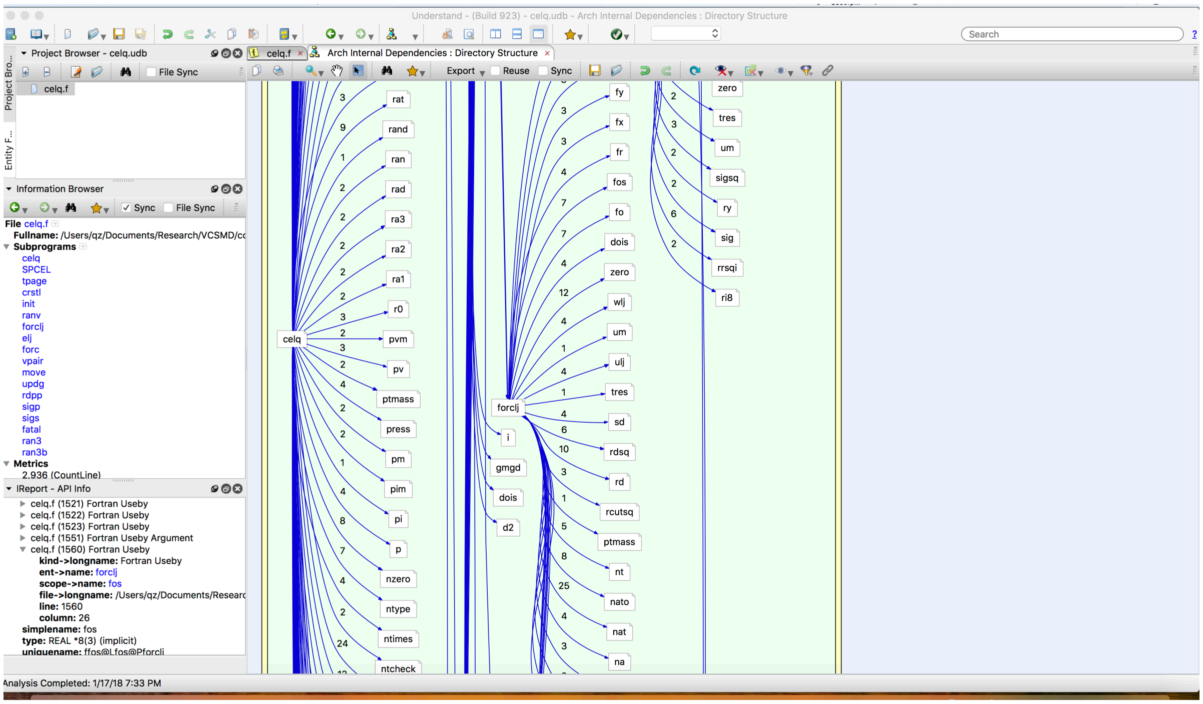
to

```
101   format(1x,4d13.5,i6)
```

i.e., any number larger than $12$. Or you will get a crowded `e` (you can try to see what I mean). I could show you how to use regular expressions to solve this problem, but the solution above is really the fastest.

And I really recommend you to use some Fortran IDE to read her code. What is unfortuante is that unlike Python, Fortran does not have an IDE like PyCharm (if you find one, please tell me). But you could try understand to analyse Fortran code. What it can do include but not limit to

1. Generate code dependency graph

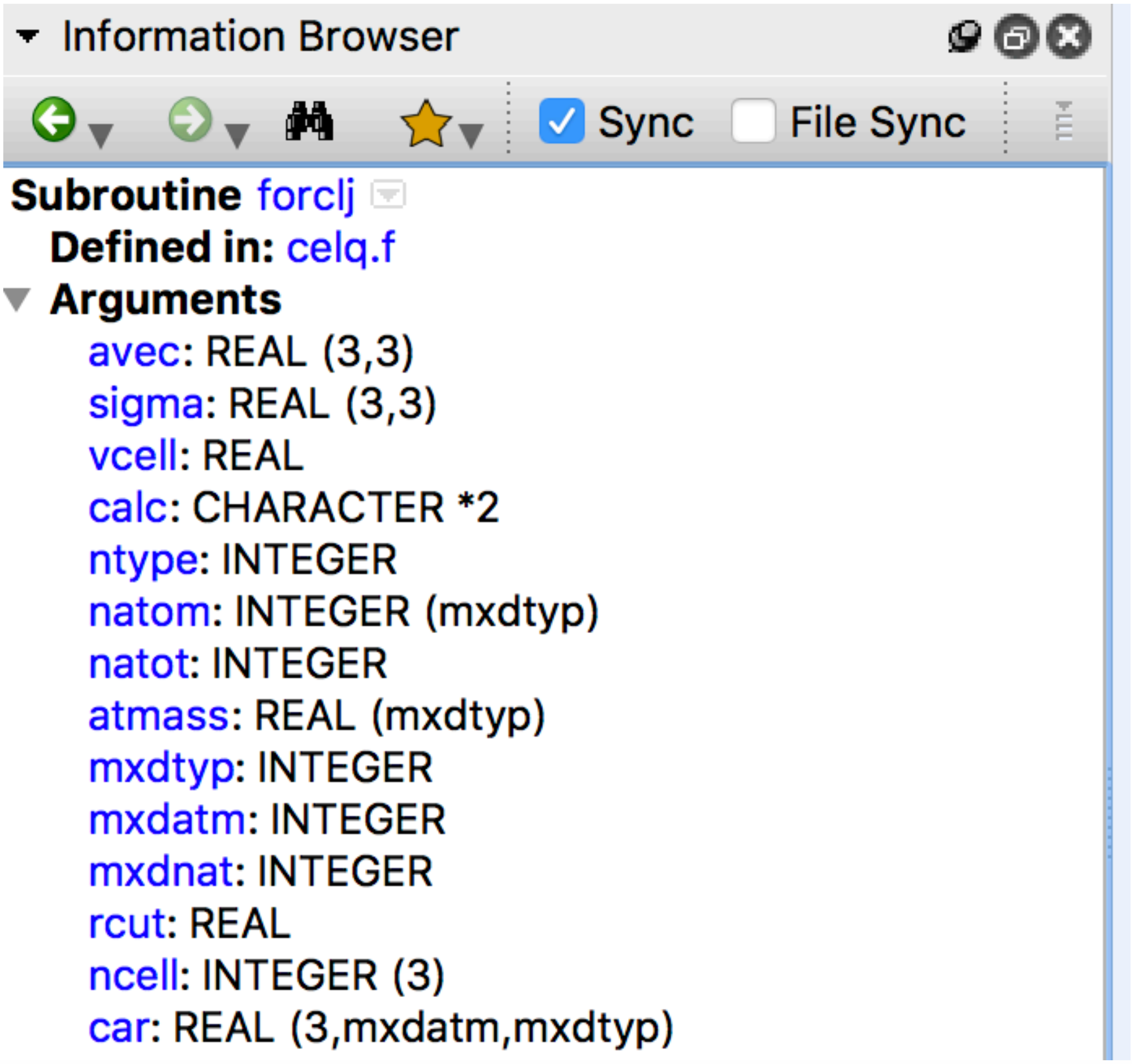


File celq.f
Fullname: /Users/qz/Documents/Research/VCSMD/cc
▼ Subprograms
celq
SPCEL
tpage
crstl
init
ranv
forclj
elj
forc
vpair
move
updg
rdpp
sigp
sigs
fatal
ran3
ran3b
▼ Metrics
2,936 (CountLine)

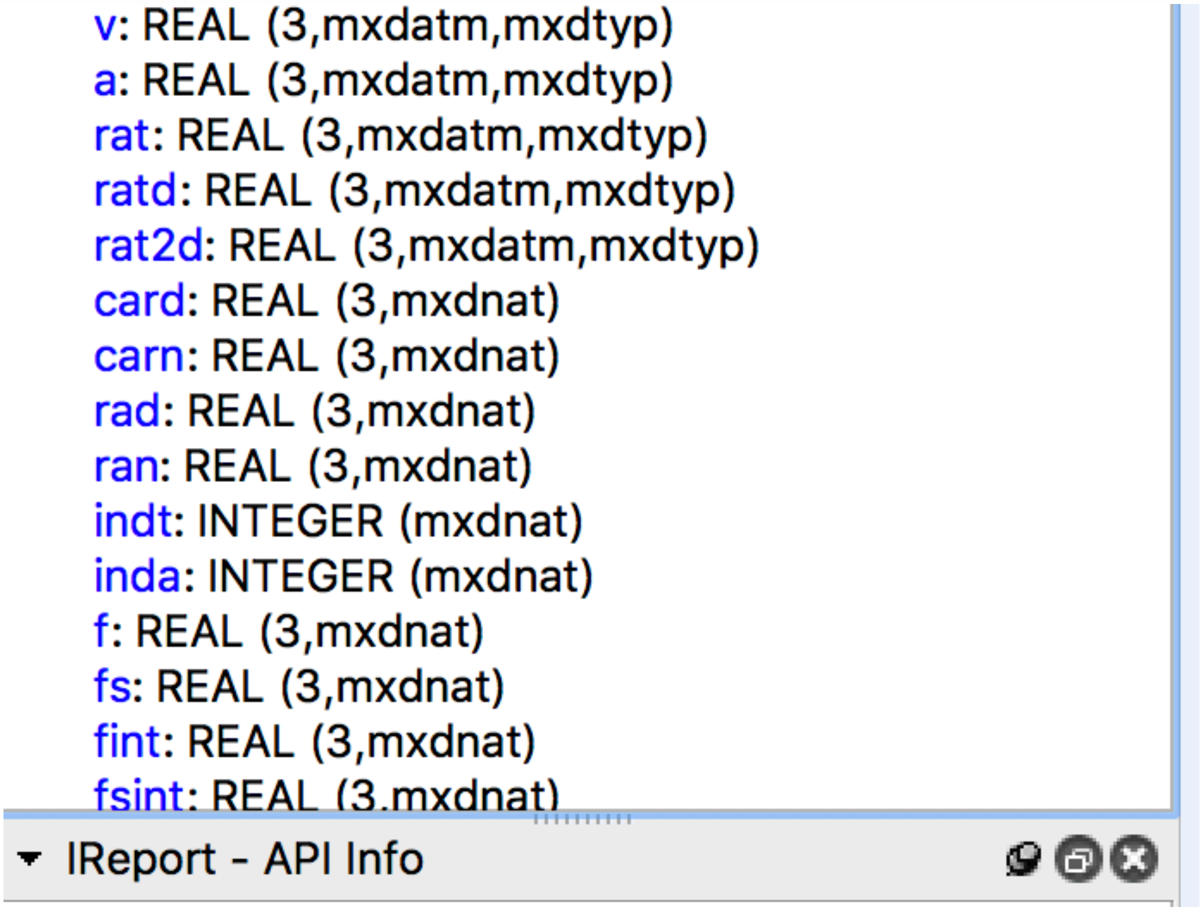
▶ celq.f (1521) Fortran Useby
▶ celq.f (1522) Fortran Useby
▶ celq.f (1523) Fortran Useby
▶ celq.f (1551) Fortran Useby Argument
▼ celq.f (1560) Fortran Useby
kind->longname: Fortran Useby
ent->name: forclj
scope->name: fos
file->longname: /Users/qz/Documents/Researc
line: 1560
column: 26
simplename: fos
type: REAL *8(3) (implicit)
uniquename: ffos@Lfos@Pforclj

Analysis Completed: 1/17/18 7:33 PM

2. Give some type information about subroutines.

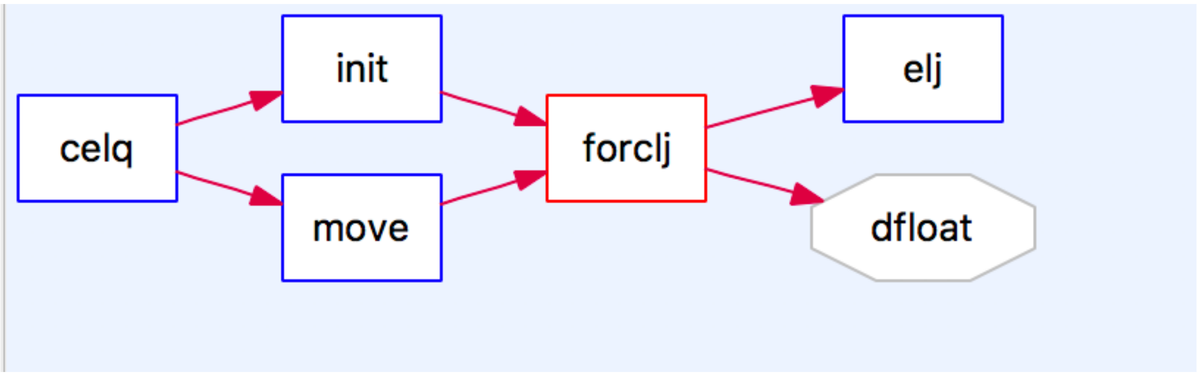


**Subroutine** forclj
**Defined in:** celq.f
▼ **Arguments**
avec: REAL (3,3)
sigma: REAL (3,3)
vcell: REAL
calc: CHARACTER *2
ntype: INTEGER
natom: INTEGER (mxdtyp)
natot: INTEGER
atmass: REAL (mxdtyp)
mxdtyp: INTEGER
mxdatm: INTEGER
mxdnat: INTEGER
rcut: REAL
ncell: INTEGER (3)
car: REAL (3,mxdatm,mxdtyp)



```
v: REAL (3,mxdatm,mxdtyp)
a: REAL (3,mxdatm,mxdtyp)
rat: REAL (3,mxdatm,mxdtyp)
ratd: REAL (3,mxdatm,mxdtyp)
rat2d: REAL (3,mxdatm,mxdtyp)
card: REAL (3,mxdnat)
carn: REAL (3,mxdnat)
rad: REAL (3,mxdnat)
ran: REAL (3,mxdnat)
indt: INTEGER (mxdnat)
inda: INTEGER (mxdnat)
f: REAL (3,mxdnat)
fs: REAL (3,mxdnat)
fint: REAL (3,mxdnat)
fsint: REAL (3,mxdnat)
```

3. Some butterfly graphs.



Anyway, this is a software which might be helpful. And you can trial use it for a year with an edu email. If you find anything better, please tell me.

Vesta can be downloaded here and I think it is easy to learn. Good luck!