2025-10-6 test system in python

We now install `testecalj` in your ecalj binary directory BINDIR.

Usage

`testecalj` uses comp.py and difnum0.py internally.
We now install `testecalj`, `comp.py` and `difnum0.py` in your bin.
(for developers: we can use `ecalj/SRC/exec/testecalj`. Then we use binaries at `ecalj/SRC/exec`.)

> testecalj [-np mpi_size] [list of tests]

Run `testecalj --help`

To run only copt and si_gwsc test with mpi_size=8, run

> testecalj -np 8 copt si_gwsc
> at ecalj/SRC/TestInstall

To run all tests,

> testecalj -np 8

The name of test directory is now `si_gwsc_work` corresponding to `si_gwsc`.

How the testecalj work?

For each test, we need finished calculations. Keep files for comparison
{such as `out.foobar, EPS*,QPU,QPD,log.*`} in the test directory such as si_gwsc/.
In the manner described in si_gwsc/test.py, we reproduce such calculations, and compare these filed. The name 'test.py' is special name used in ecalj/SRC/exec/testecalj.

PROF

How to add your test

If you like to add new test, you do run calculations
and keep some files for comparison. Then you add your own test as in si_gwsc/test.py.

```
from comp import runprogs,diffnum,dqpu
def test(args,bindir,testdir,workdir):
        gwsc0= bindir + f'/gwsc 0 -np {args.np} '
        tall=''
        runprogs([
                 "rm log.si QPU",
                 gwsc0+ " si"
        ])
        dfile="QPU"
        for outfile in dfile.split():  #this loop is for dfile="QPU QPD"
                tall+=dqpu(testdir+'/'+outfile, workdir+'/'+outfile)
        outfile='log.si'
        tall+=diffnum(testdir+'/'+outfile, workdir+'/'+outfile,tol=3e-
3,comparekeys=['fp evl'])
        return tall
```

This is called from `testecalj`. As this shows, we need to

1. Speficy commnands for test.
2. Write steps of computation in runprogs.
3. Comparison (diffnum is for numerical comparison for lines including 'fp' and 'eval') in this case.

See other SRC/TestInstall/*/test.py as examples.