

Precision exercise

```
msub -X -q debug -I -l partition=t1,size=32,walltime=02:00:00
cd $CSCRATCH/$USER
mkdir DDT
cd DDT
tar -xf ~William.Cooke/ddt_examples.tar
```

```
cd precision
```

Inspect the code precision.F90

Exercise

1) A real*4 (32bit) example.

```
./run_precision_r8.csh
```

This should add $1e-N$ in the N loop

i.e. the printout should be 1.0 1.10 1.110 1.1110 etc.

What does the printout show?

What happens near the end of the loop?

2) A real*4 (32bit) example.

```
./run_precision.csh
```

DDT Exercise

```
cd cstartmpi
```

```
module load ddt
```

Exercise 1)

```
./run_cstartmpi.csh
```

This sets up the executable

```
aprun -n 4 ./cstartmpi.exe
```

```
aprun -n 4 ./cstartmpi.exe foo bar
```

"Rank 0 has 3 arguments."

```
aprun -n 4 ./cstartmpi.exe foo bar blah
```

"Rank 0 has 4 arguments."

```
ddt -n 4 ./cstartmpi.exe foo bar blah
```

At the opening window, edit the argument list if you wish.

Press Run

When you get to the source window, take a look around at the different windows available.

Press Play (Green arrow button under Session)

A SIGSEGV error message window will pop up.

Press Pause.

Look at the "Current Stack" window.

Click on the top element "#4 main..."

Click on the "Current Line(s)" tab. What is the value of y?

Is the value of y correct?

Is there anything wrong with this loop?

Correct the loop and run ./run_cstartmpi.csh again to recompile the code.

Session-> New Session-> Run
Run the code to completion

Exercise 2)

There is another problem with the code. It segfaults at higher core counts.
Try to figure out why.

Session-> New Session-> Run
Change the core count to > 4

Find out what the problem is.

TRI-DIAGONAL SOLVER EXERCISE.

```
cd trisol
./run_trisol.csh
```

```
> aprun -n 4 ./trisol.exe
*** Solution correct
|x| / (sqrt(n)*epsilon*(|A|*|x| + |b|) = 3.7352-215
```

```
> ddt -n 4 ./trisol.exe
Runs to completion.
So everything is good?? No. There is memory error in here.
```

Let's turn on some memory debugging options.

Restart the session. (Session-> New Session-> Run)
Check the Memory Debugging box.
Click on Details.

Check Preload the memory.
In the Heap debugging options check the High box.
Leave the Heap Overflow unchecked (for now).

Run the program.
DDT reports an array overflow.

Where is the overflow?
We'll use guard pages to find out.

Restart the session.
Click on Memory Debugging->Details.

Check the "Add guard pages" box under "Heap Overflow/Underflow Detection"
Leave at "1" and "After"

Run the program again.
DDT reports a memory error in check.f90
Click Pause
Look at the size of res in the "Current Lines" tab.

res is size 4095
k = 4095 so we are trying to write beyond the end of res

Exercise 3
Fix the code by inserting
DO k = 1, block_size
 res(k) = k
END DO

And run ./run_trisol.csh to recompile
Then
ddt -n 4 ./trisol.exe

You should get a Memory Error in matnrm.
Pause the job and examine values of i, k, j1, j2 on different processors.
You'll need to pause all cores.

See if you can figure out what is wrong.

Hints are in the presentation.