De-bugging Tips

7.4 Symptoms and probable causes

Have you got rounding errors?

- Don't do floating point calculations using integers. Make sure your precision is consistent and adequate. Make sure the precision of the right hand side of an equation matches the type of variable you are assigning to.

Are your calculations completely wrong?

- Initialise all your variables – don't forget arrays!
- Make sure your arrays are big enough to hold all the data.
- Check that arguments passed to subroutines agree exactly in size, type and position
- Is the program's logic working the way it should?

7.5 Wise precautions and time saving tips

- You must **not** test floating point numbers for equality. Example:
  `if (x == 1) then...`

does not work.

- Should you be using the absolute value of a calculation? Example:
  `if (abs(x-y)<.00001) then`
- Don't have overly elaborate logical tests. It's probably better to test one or two things at a time rather than this sort of thing…
  `if (((x.AND.y).OR.z > 10).OR.(.NOT. xx < 0)) then …`

you might think you understood it when you wrote it, but imagine trying to figure out what's happening if the program breaks!

- **Don't try and write a complicated program all at once. Write it a piece at a time and check that each piece is working correctly before doing the next bit.**
- Use 'implicit none' at the start of all programs and subroutines.
- If your program needs data to be entered by the user, you will save hours of time by taking the trouble to read in the data from a file while the program is in development.
- Always do a test for 'division by zero' when dividing.
- **BE NEAT!** Good programming is like plain speaking – there's no mileage in tricky, difficult to read code.

7.6 How to find the bugs

Use a print statement to print out the values within the program – take a look at this code…

```
x  = x + 1
z  = x * y
print  *, 'debug statement 1 value of x,y,z', x,y,z
do  ii  =1,10
   z = x *  ii
   if (ii == 5) then
     print *, 'debug do loop value of z when ii =  5',z
   end if
end do
if (z>2000) then
   print *, 'debug statement – z>2000  value of z ',z
   stop
end if
```

Notice how we can print out values of specific variables, stopping the program if necessary.

Index