One dimensional Scattering

Josh Fernandes
Department of Physics
California State University Long Beach

January 29, 2014

Abstract

The results of a one dimensional scattering equation are plotted.

1 Introduction

introduction goes here.

2 The Fortran95 code

The fortran code solves a one dimensional scattering equation. In our first Fortran95 code, let us consider the path of a ball thrown in the air from height h with initial velocity $\vec{v}_0$ in the direction which makes α angle with the horizontal.

The Makefile in Fig. 1 describes the structure of the code. The codes in `OBJS1` are compiled together. The `.o` files were created from the `.f95` file by using the `F95` command with flags `F95FLAGS`. In building the executable code `LDFLAGS` were used, which contains the `LIB` library. These `LDFLAGS` are optimal for Intel Core2 architecture and uses the Apple's `vecLib` library from Xcode. By typing `make` we can produce the executable file.

Basically we code the formula $x = v_0 t$ and

$$T = \frac{\frac{1}{4}\Gamma^2}{(E-E_0)^2 + \frac{1}{4}\Gamma^2}. \tag{1}$$

Listing 1: Typical `Makefile`

```
OBJS = numtype.o scattering.o

PROG = scattering

F95 = gfortran

F95FLAGS = -O3 -funroll-loops -fexternal-blas

LIBS = -framework vecLib

LDFLAGS = $(LIBS)

all: $(PROG)

$(PROG): $(OBJS)
    $(F95) $(LDFLAGS) -o $@ $(OBJS)

clean:
    rm -f $(PROG) *.{o,mod}

.SUFFIXES: $(SUFFIXES) .f95

.f95.o:
    $(F95) $(F95FLAGS) -c $<
```

Our first Fortran95 code is shown below. The precision is defined in the Module `NumType`. The constants parameters are defined in Module `setup`. The implicit none statement ensures that all the variables has to be defined. We use `do-end do` structure for the loop, and exit if the next height would be negative.

Listing 2: Module `NumType`

```
module NumType

    save
    integer, parameter :: dp = kind(1.d0)
    real(dp), parameter :: pi = 4*atan(1._dp)
```

```
    complex(dp), parameter :: iic = (0._dp,1._dp)

end module NumType
```

Listing 3: The `setup` module.

```

module setup

    use NumType
    implicit none
    real(dp), parameter :: g = 10._dp
    real(dp), parameter :: h0 = 20._dp
    real(dp), parameter :: v0 = 15._dp
    real(dp), parameter :: alpha = 30*(pi/180)

end module setup
```

Listing 4: The `scattering.f95`

```


module setup

    use NumType
    implicit none
    real(dp), parameter :: g = 9.81_dp
    real(dp) :: h0, v0, alpha


end module setup

program scattering_path

    use setup
    implicit none
    real(dp) :: E, dE, E0, gamma(4)
    REAL, DIMENSION(:, :), ALLOCATABLE :: T
    integer :: i, j, N, M, DeAllocateStatus,
    AllocateStatus, Emax

```

```fortran

    gamma  = (/ 1.0, 0.5, 0.1, 0.01/)

    E = 0._dp
    dE = 0.01_dp
    Emax = 10_dp

    E0 = 5_dp
    j = 0._dp

    N = size(gamma)
    M = (Emax/dE)+1

    ALLOCATE ( T(N, M), STAT = AllocateStatus)
    IF (AllocateStatus /= 0) STOP "*** Not enough memory ***"

    do while (E < Emax)
        j = j + 1
        do i=1,4
            T(i,j) = (.25*(gamma(i)**2))/((E-E0)**2 + .25*(gamma(i)**2))
        end do
        print *, E, T(1,j), T(2,j), T(3,j), T(4,j)
        E = E + DE
    end do

    DEALLOCATE (T, STAT = DeAllocateStatus)
end program scattering_path
```

We can run the code by typing `ball`. The results are printed on the screen, or we can redirect to the file `1` by typing `ball > 1`. The `Plot` provides the picture in Fig. 1, which, with little work, can be put in a much nicer form.

3 Summary and conclusions

We presented our first Fortran95 code [1]. We plotted the height of a falling ball as a function of time. The numerical results agree with our prior knowledge and expectations.

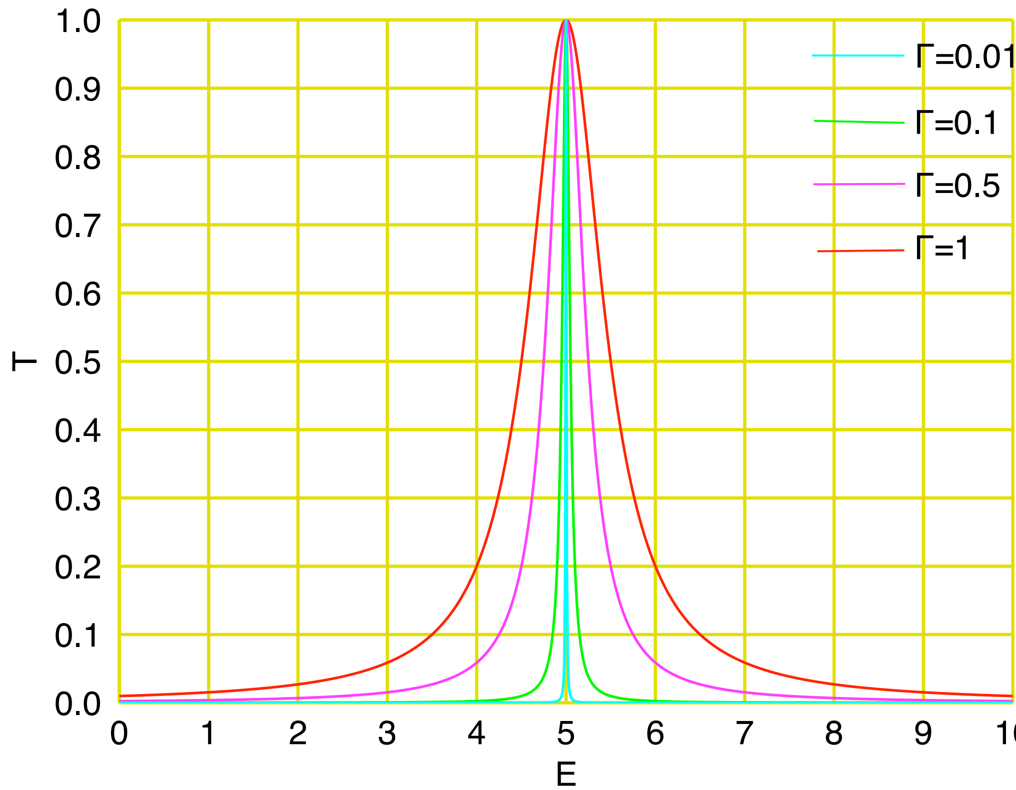

Figure 1: Results of the `scattering.f95` code plotted on a linear scale.

References

[1] M. Metcalf, J. Reid and M. Cohen, *Fortran 95/2003 explained*. Oxford University Press, 2004.