Physics 562 - Computational Physics

Assignment 1: Fourier's law

Josh Fernandes
Department of Physics & Astronomy
California State University Long Beach

February 3, 2014

Abstract

For this assignments we are practicing some pasic coding concepts in Fortran as well as LaTeX We are given Fourier's law to paramaterize and then graph the outcome Θ. We then practice basic unix command to pipe our data in to a data file and graph using PlotPublish. We then needed to publish our results in /LaTex and learn to use Makefiles.

1 Introduction

This is the first assignment for the class. The purpose of the assignment is to familiarize ourselves with Fortran,Make file, and LaTeX notation. Fortran dates back to 1953, but it has been updated nine times since it's release. This class focuses on Fortran95, but there are more recent versions of Fortran with another update coming in 2015. This paper solves a scattering coefficient equation for several values of Γ. The results are plotted on the same graph, so the effect of Γ on the shape of the curve can be determined.

2 The Fortran95 code

The fortran code solves a one dimensional scattering equation. There is a transmission coefficient that represents a probability of a particle being transmitted. It ranges from 0 to 1. It is centered at some Energy E_0 . The function also depends on a constant denoted by Γ. The formula is

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

The Makefile in Listing 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.

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 $<
```

The 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. `do while-end do` is used to increase the value of E while also calculating T for the four different Γ values.

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 `scattering.f95`

```


module setup

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


end module setup

program scattering_path

    use setup
    implicit none

    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
```

The code is run by typing `./scattering`. The results are printed on the screen, or the data can redirect to the file `1` by typing `./scattering > 1`. The `Plot` provides the picture in Fig. 1 and Fig. 2.

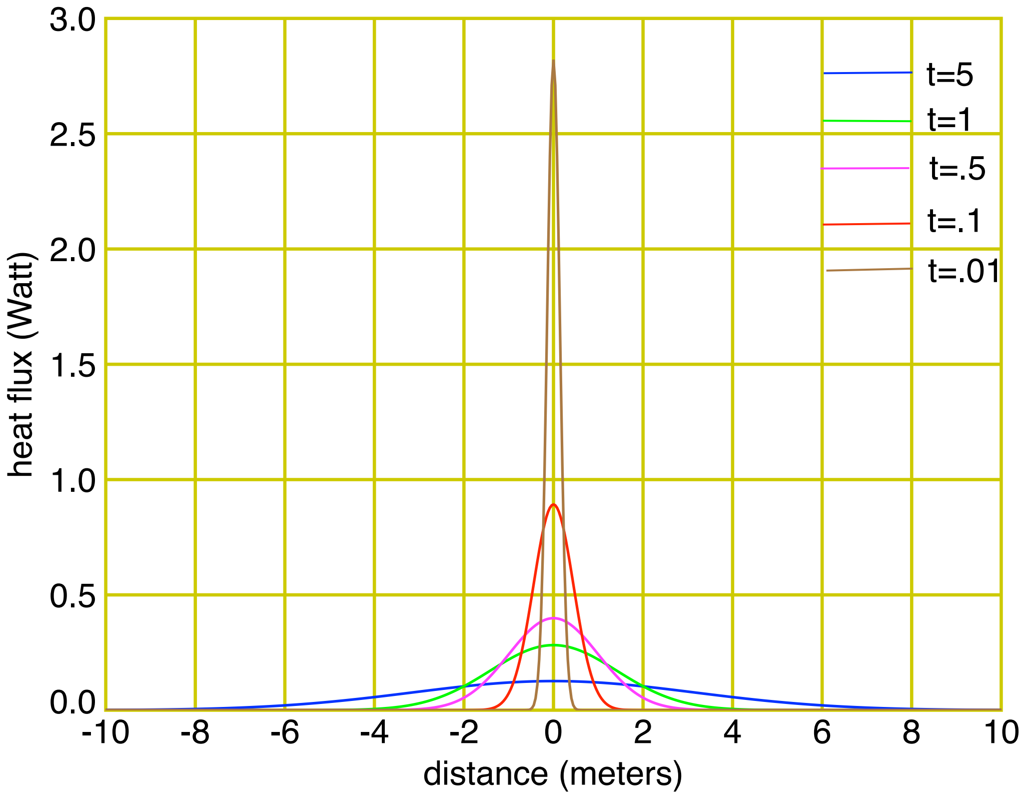

Figure 1: Results of the `heat.f95` code plotted on a linear scale.

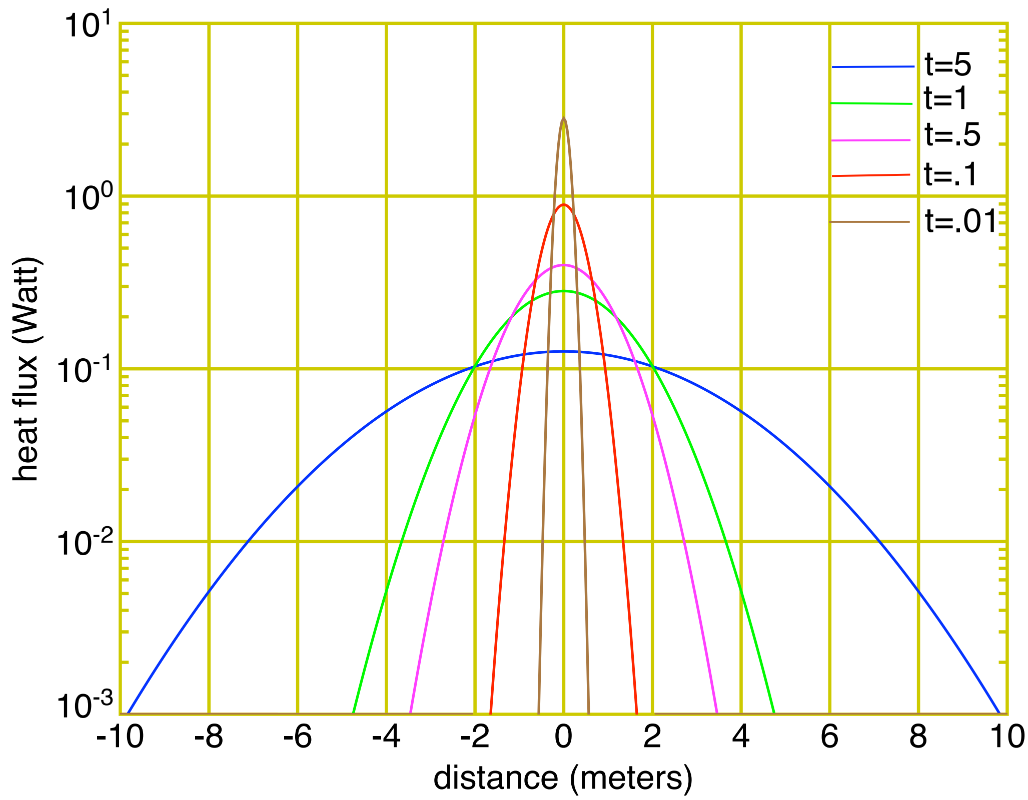

Figure 2: Results of the `heat.f95` code plotted on a log scale.

3 Summary and conclusions

The first Fortran95 code [1] for the class was written. The transmission coefficient as a function of E with four different Γ values was plotted. The numerical results agree with our prior knowledge and expectations.

References

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