My Thesis

Joe Science

Department of Physics and Astronomy

University of Leeds

A thesis submitted for the degree of

Doctor of Philosophy

30th January 3000

This thesis is dedicated to...

Acknowledgements

My thanks to...

Abstract

My abstract in here...

Abbreviations

k_B	Boltzmann's constant
k_BT	Thermal energy
...	...

Contents

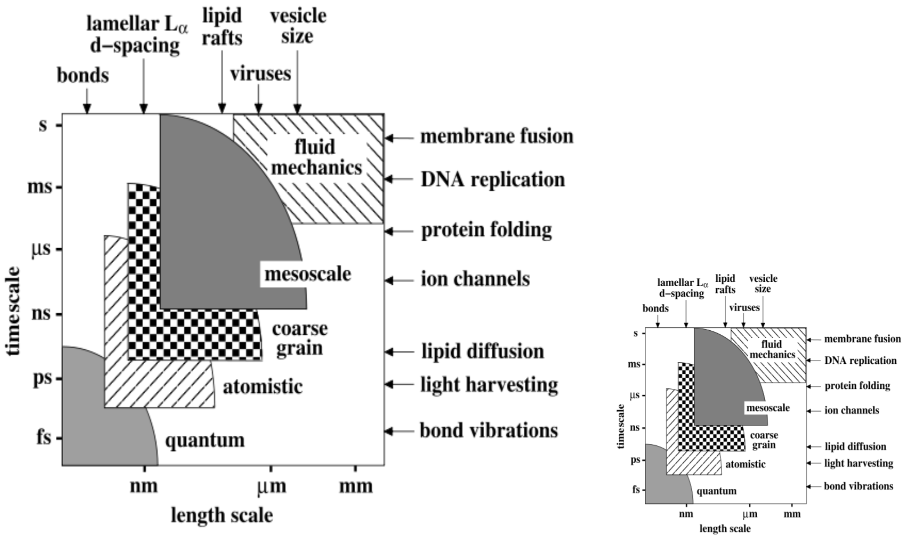
List of Figures

Chapter 1

Introduction

1.1 Introduction

Ever advancing developments in computational power....



(a) (b)

Figure 1.1: Simulation Scale Cartoon (Nielsen *et al.*, 2004).

1.1.1 Background

Appendix A

Code samples

A.1 Random Number Generator

The Bayes Durham Shuffle ensures that the psuedo random numbers used in the simulation are further shuffled, ensuring minimal correlation between subsequent random outputs (Press *et al.*, 1992).

```
#define IM1 2147483563
#define IM2 2147483399
#define AM (1.0/IM1)
#define IMM1 (IM1-1)
#define IA1 40014
#define IA2 40692
#define IQ1 53668
#define IQ2 52774
#define IR1 12211
#define IR2 3791
#define NTAB 32
#define NDIV (1+IMM1/NTAB)
#define EPS 1.2e-7
#define RNMX (1.0 - EPS)

double ran2(long *idum)
{
```

```
/*---------------------------------------------------*/
/* Minimum Standard Random Number Generator          */
/* Taken from Numerical recipies in C                */
/* Based on Park and Miller with Bays Durham Shuffle */
/* Coupled Schrage methods for extra periodicity     */
/* Always call with negative number to initialise    */
/*---------------------------------------------------*/

int j;
long k;
static long idum2=123456789;
static long iy=0;
static long iv[NTAB];
double temp;

if (*idum <=0)
{
  if (-(*idum) < 1)
  {
    *idum = 1;
  }else
  {
    *idum = -(*idum);
  }
  idum2=(*idum);
  for (j=NTAB+7;j>=0;j--)
  {
    k = (*idum)/IQ1;
    *idum = IA1 *(*idum-k*IQ1) - IR1*k;
    if (*idum < 0)
    {
      *idum += IM1;
    }
    if (j < NTAB)
```

```
      {
        iv[j] = *idum;
      }
    }
    iy = iv[0];
  }
  k = (*idum)/IQ1;
  *idum = IA1*(*idum-k*IQ1) - IR1*k;
  if (*idum < 0)
  {
    *idum += IM1;
  }
  k = (idum2)/IQ2;
  idum2 = IA2*(idum2-k*IQ2) - IR2*k;
  if (idum2 < 0)
  {
    idum2 += IM2;
  }
  j = iy/NDIV;
  iy=iv[j] - idum2;
  iv[j] = *idum;
  if (iy < 1)
  {
    iy += IMM1;
  }
  if ((temp=AM*iy) > RNMX)
  {
    return RNMX;
  }else
  {
    return temp;
  }
}
```

References

NIELSEN, S., LOPEZ, C., SRINIVAS, G. & KLEIN, M. (2004). Coarse grain models and the computer simulation of soft materials. *J. Phys. Condens. Matter*, **16**, R481–R512. 7, 1

PRESS, W. *et al.* (1992). *Numerical recipes in C*. Cambridge University Press Cambridge. 2