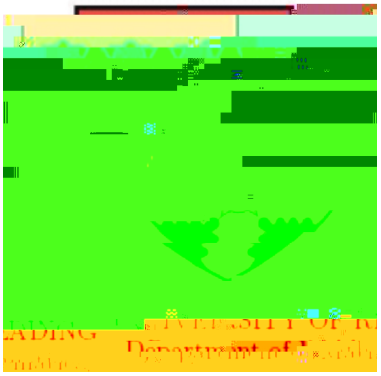


The University of Reading
The Department of Mathematics and Statistics
MSc Dissertation

SELF - CONSISTENT FIELD CALCULATIONS ON A VARIABLE RESOLUTION GRID

Author:
Melios Michael

Supervisors:
Professor Michael J. Baines
Professor Mark W. Matsen

August 2011
This dissertation is submitted to the Department of Mathematics in partial ful lment of the requirements for the degree of Master of Science

Declaration

I con rm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Signed .....

Abstract

The Self-Consistent Field Theory, otherwise known as Mean Field Theory, represents interactions by two static fields acting on two polymer segments (A, B). The model corresponds to a melt of AB diblock copolymers subject to a local incompressibility constraint. Exploiting the simplest classical microstructure (Lamellar) the density distributions of the copolymer blocks are computed by applying the Crank-Nicolson algorithm on a uniform mesh. The aim of this dissertation is to increase the numerical efficiency of the calculations by employing an adaptive mesh using subdivision. The dissertation contains a study of the effectiveness of using the h refinement technique on the computation of the copolymer propagators with the equation $\frac{\partial}{\partial s}q(\mathbf{r};s) = \frac{a^2N}{6}\nabla^2 q(\mathbf{r};s) - w(\mathbf{r};s)q(\mathbf{r};s)$. Preliminary studies are carried out using a uniform mesh method and globally refining the mesh before subdividing the mesh in areas of interest. Numerical results of the total partition function and the segment concentration distributions are compared and conclusions drawn on the space size step and local refinement factors.

Acknowledgements

I would like to thank all the sta of the Mathematics Department for being friendly and supportive throughout the duration of my studies. Particular thanks must go to Dr. Peter Sweby and Sue Davis for all their help and for their spontaneous replies and guidance. I would also like to thank Prof. Mark Matsen for his help and supervision in my dissertation. Particular thanks for his help, guidance, conversations, motivation and positivity in all matters and situations whether they re ected on the course or general topics and personal problems must go towards Prof. Mike Baines. Your supervision, ideas, jokes and patience have really been appreciated and inspiring.

Lastly I would like to thank my family and close friends for their understanding and support which made the completion of the course and dissertation easier and possible.

Contents

1 Introduction 5
1.1 Background 5
1.2 Reasons for Research 6
1.3 The Model 6
1.4 Aim of this study 9
1.5 Adaptive Methods 9

2 Preliminary Stage 11

Chapter 1

Introduction

1.1 Background

A polymer can be de ned as a macromolecule. It is essentially constructed from several repeated monomer building blocks or chemical units, linked together into one or more chains [4]. Typically there could be several hundred to several thousand monomer units in a polymer. When the monomers that build a chain are all of the same chemical structure, in other words if they are identical, the polymer is called a Homopolymer. If however it involves two or more chemical distinct monomers then, the result is termed a copolymer. In our current investigation, we will be considering block copolymers, these refer to molecules grouped together as blocks. A linear diblock copolymer is a polymer constructed by attaching one end of a linear homopolymer of one type of chemical units to the end of another linear homopolymer of distinct types, creating a longer but still linear molecule. Symbolically we denote A and B the two distinct types of chemical units or monomer and the portion of the chain which is of type A is referred to as the A block, and similarly the porton of the chain which is of type B is referred to as the B block. Thus, diblock copolymer refers to a copolymer comprised of two blocks of distinct species [6].

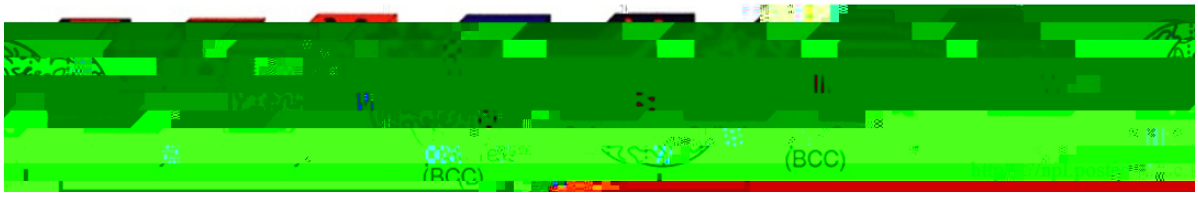


Figure 1.1: Classical Microstructures

represents interactions by two static elds acting on A,B segments. The model corresponds to a melt of AB diblock copolymers subject to a local incompressibility constraint. It uses the applicability of the Gaussian model and develops the necessary statistical mechanics for a single chain which is subject to an external eld, $w(\mathrm{r})$. According to the Gaussian model, diblock copolymers are treated as microscopic elastic threats [14].

We should de ne the following notation that will be used in our model:

r_i = i^{th} monomer

r (s) = a function explaining the coarse-grained trajectory of the polymer

s = a parameter indicating the interval of the chain (segment) and is de ned within $0 \quad s \quad 1$

subscript is ued to label di erent molecules

N total number of segments

When discretising the partial di erential equation 1.2, s is treated similarly to a time variable and r as the space variable.

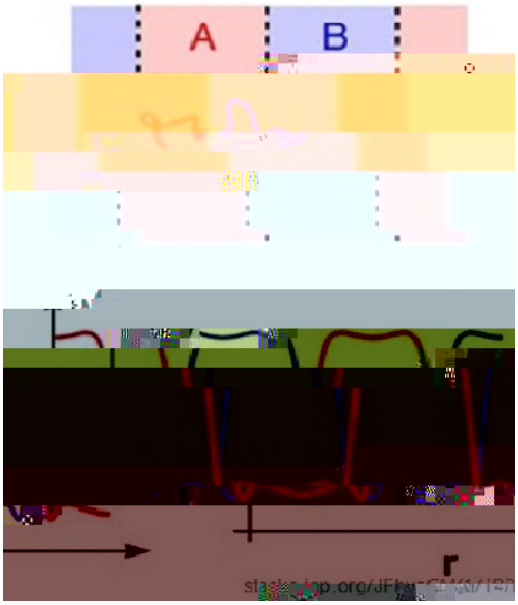
Each molecule is parametrised by a variable s that increases from 0 to 1 along the length. The partition function Q for a single copolymer experiencing chemical potentials q and q that exerts forces, respectively, on the A and B blocks is given by

$$Q = \int q(\mathrm{r};s)q\ (\mathrm{r};s)dr \tag{1.1}$$

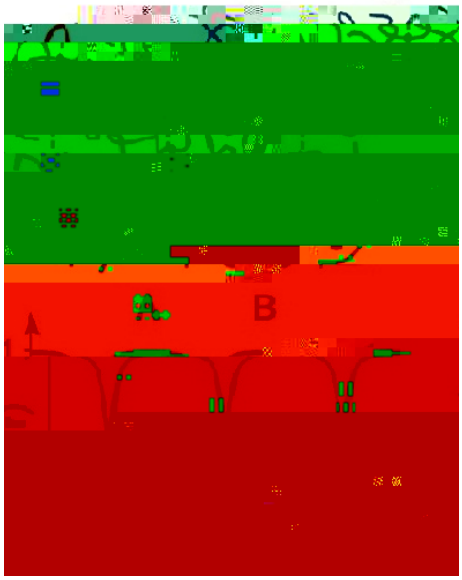
where the copolymer propagator satis es

$$\frac{@}{@s}q(\mathrm{r};s) = \frac{a^2N}{6}\, r^{\,2}q(\mathrm{r};s) \quad w(\mathrm{r};s)q(\mathrm{r};s) \tag{1.2}$$

and can be evaluated starting from $q(r;0) = 1$. Similarly for $q\ (r;s)$ the di erential (1



(a) External Fields



(b) Segment Concentration Dist.

Figure 1.2: [8]

and starting from $q(r;1) = 1$. The function $w(r;s)$is given by

$$w(r;s) = \begin{cases} w_A(r); & 0 \quad s \quad f \\ w_B(r); & f \quad s \quad 1 \end{cases} \tag{1.4}$$

For our case, a symmetric diblock copolymer melt, $f = \frac{1}{2}$, the model exhibits a microphase phase separation into a lamellar phase for segregation strengths N, where is the Flory parameter that quanti es the repulsive interaction of the chemical units. The boundary conditions for this problem can either be periodic or re ective Neumann conditions [1] .

$_A(r)$ and $_B(r)$ correspond to ensemble-averaged segment concentration distributions and from the schematic diagram in Figure 1.2b we can see the concentrations of A and B segments at di erent points r.

The incompressibility assumption is then given by $_A + _B = 1$. The segment concentrations are de ned as follows:

$$_A(r) = \frac{V}{Q} \int_0^f q(r;s)q\ (r;s)ds \qquad _B(r) = \frac{V}{Q} \int_f^1 q(r;s)q\ (r;s)ds \tag{1.5}$$

"Re ning indicators" are often used to identify portions of the domain in need of additional resolution [10].

This dissertation will centre around the use of the local grid re nement. The aim of the h type adaptive procedure is to achieve a higher rate of convergence and thus reach the desired accuracy with minimal cost. Given that we have su cient information describing where best to re ne the mesh, then there are several ways of approach. Using h-re nement, we obtain a new mesh by simply subdiving the intervals.

Using this method we can break the mesh into smaller pieces when necessary and even coarsen the mesh where the solution is very smooth. However, large jumps in

Chapter 2

Preliminary Stage

We are interested in obtaining the solution to the Self-Consistent Field equations in which the normalised density distributions, (r) re ect a lamellar morphology. In this symmetry, (r) varies only along one axis, and exhibits a periodic variation. (r) can thus be reduced to a function of only one co-ordinate, *r*, and be periodic.

The nite di erences method will be used to approximate the solution of the di usion equation 2.1. The basic idea of the nite di erence method of solving PDEs is to replace spatial and time derivatives by suitable approximations, then to numerically solve the resulting di erence equations. Speci cally instead of solving $q(r;s)$ with r continuous, we solve $q_{i;j}$ $q(r_i;s_j)$ where r_i i x and s_j j s. We will have a grid similar to Figure 2.1

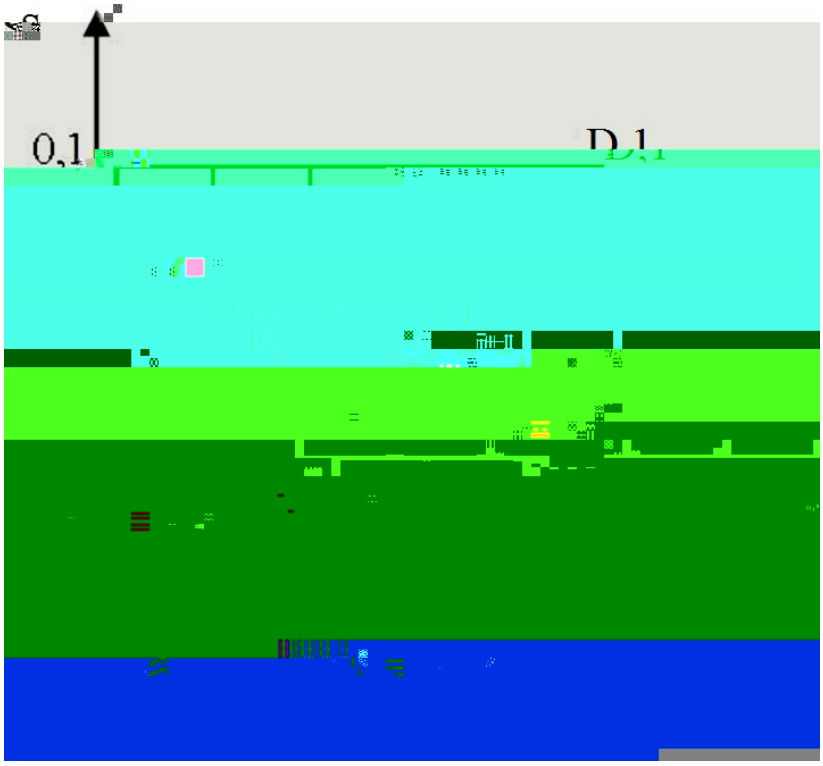


Figure 2.1: Grid

$$\frac{@q}{@r} = \lim_{r \to 0} \frac{q}{r} \tag{2.3}$$

the derivatives evaluated at the grid points $(r; s) = (r_i; s_j)$ can be approximated in many di erent ways, the simplest being the following:

and dr the size of the space step

D will denote the total length of space, 0 s D

t_{max} the total length of time and as previously mentioned, the segments s behave similarly to the time variable so $t_{max} = 1$

N_t the total number of time steps and ds the size of time step

nally $r(i) = (i \quad 1) \quad dr$ where $dr = \frac{D}{N}$

approaches zero in the limit that $s \to 0$ and $r \to 0$. The scheme is found to be "consistent", first order in time and second order in space.

In order to guarantee that the scheme will give a good approximation to the true solution of the diffusion equation, when the discretised equation approaches the exact solution then the numerical scheme is termed convergent.

For a linear solution such as the diffusion equation, convergence is dependent on the stability of the numerical scheme, it is termed stable if the amplification factor remains bounded during calculations. According to the *Lax Equivalence theorem* schemes that are convergent are those that are consistent and stable [7].

Therefore, for a properly posed initial value problem for a linear PDE and a consistent finite difference approximation, stability is the necessary and sufficient condition of convergence. The explicit scheme 2.12 is stable and therefore convergent when

$$\frac{s}{(\ r)^2} \quad \frac{1}{2} \tag{2.13}$$

The main advantage of the explicit scheme is that it's easy to solve numerically. However, the stability condition raises issues as we are bound by the physical problem on having total space length $D = 2.336784$ and total time variable $t_{max} = 1$ and thus restricting us on the choice of total number of space and time steps $N_x; N_t$. We can therefore conclude that the stability condition and the scheme's first order in time truncation error restrict the accuracy of our numerical result.

2.2.1 Applying the Explicit Scheme

Since the explicit Scheme is relatively easy to solve numerically, we will use a FORTRAN program that will numerically solve the partial differential equation 2.1 that defines the copolymer propagators.

Using the finite difference method as shown in the previous section we can obtain the discretised equation of 2.1,

$$q_{i;j+1} = q_{i;j} + \frac{1}{6}\frac{ds}{(dr)^2}(q_{i+1;j} \quad 2q_{i;j} + q_{i\ 1;j}) \quad w_j q_{i;j} \tag{2.14}$$

For the q the PDE is identical to 2.1 but with the right hand side multiplied by -1. Therefore the discretised form is very similar but with a small difference as shown below,

$$q_{i;\mathbf{j-1}} = q_{i;j} + \frac{1}{6}\frac{ds}{(dr)^2}(q_{i+1;j} \quad 2q_{i;j} + q_{i\ 1;j}) \quad w_j q_{i;j} \tag{2.15}$$

In both cases, equations 2.14 and 2.15, we use reflective boundary conditions

which are de ned in the program as, $q_{0;j} = q_{2;j}$ and $q_{N_x+1;j} = q_{N_x \ 1;j}$. They are de ned similarly for q which is denoted as q_1 in the program. The initial data for the q propagator is $q(r;0) = 1$ and for the q is $q\ (r;N_t) = 1$.

The function w_j, the ev5.619 4.338 Td [(e Td [1(similarly)-hrly)-h4vd,.842 0 Tdd [1(gi4vd,1)28

In Figure 2.4 the blue line denotes w_A and the green line w_B. The external fields have been computed using 15 space steps and 80 time steps for 0 r 2:336784 and 0 s 1 and because there are only 15 space steps the plot is not smooth nor is it accurate enough. We therefore try plotting it using N_x = 100 and N_t = 3592,

Figure 2.5: Program - Stability condition noti cation

The numerical solution of q when choosing 15 space steps and 80 time steps has been plotted in MATLAB and is shown in Figure 2.6. The results could also be plotted on a logarithmic scale so that the plots show more details in all the domain and the propagator's behaviour, however the propagator plots are only shown for reference. The aim is to compute the segment concentrations with great accuracy and to meet the incompressibility condition

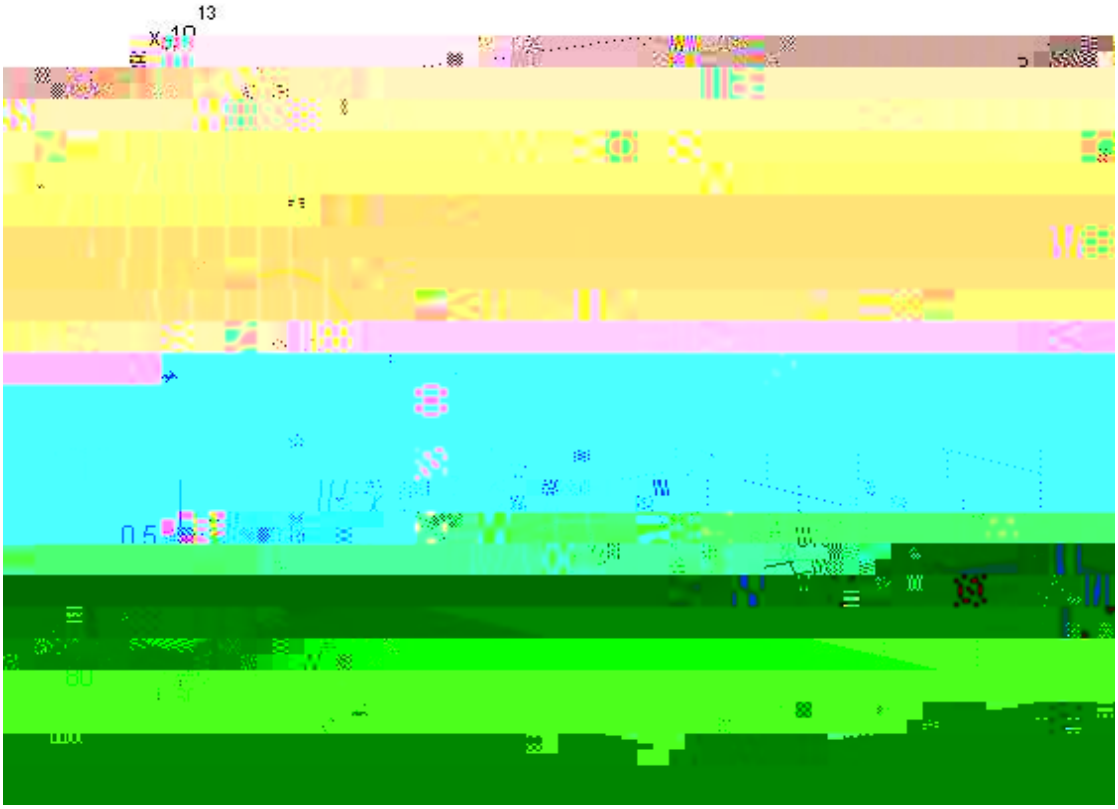


Figure 2.6: q propagator - explicit scheme - $N_x = 15$, $N_t = 80$

In order to check the correctness of the plot, a small test we could perform is to remove the $w_{;i}$ function, the external fields, from the partial differential equation 2.14. This would transform our PDE in a pure diffusion equation and in fact we would be solving the Heat Equation $\frac{@q}{@s} = \frac{@^2q}{@r^2}$, of which the expected result is known. The discretised form of our PDE would then look like equation 2.18. The fact that our initial data is equal to 1 throughout the whole domain including the boundary points, would lead us to expect a constant value of 1 as a solution and a flat plot 2.7

$$q_{i;j+1} = q_{i;j} + \frac{1}{6}\frac{ds}{(dr)^2}(q_{i+1;j} \quad 2q_{i;j} + q_{i \ 1;j}) \tag{2.18}$$

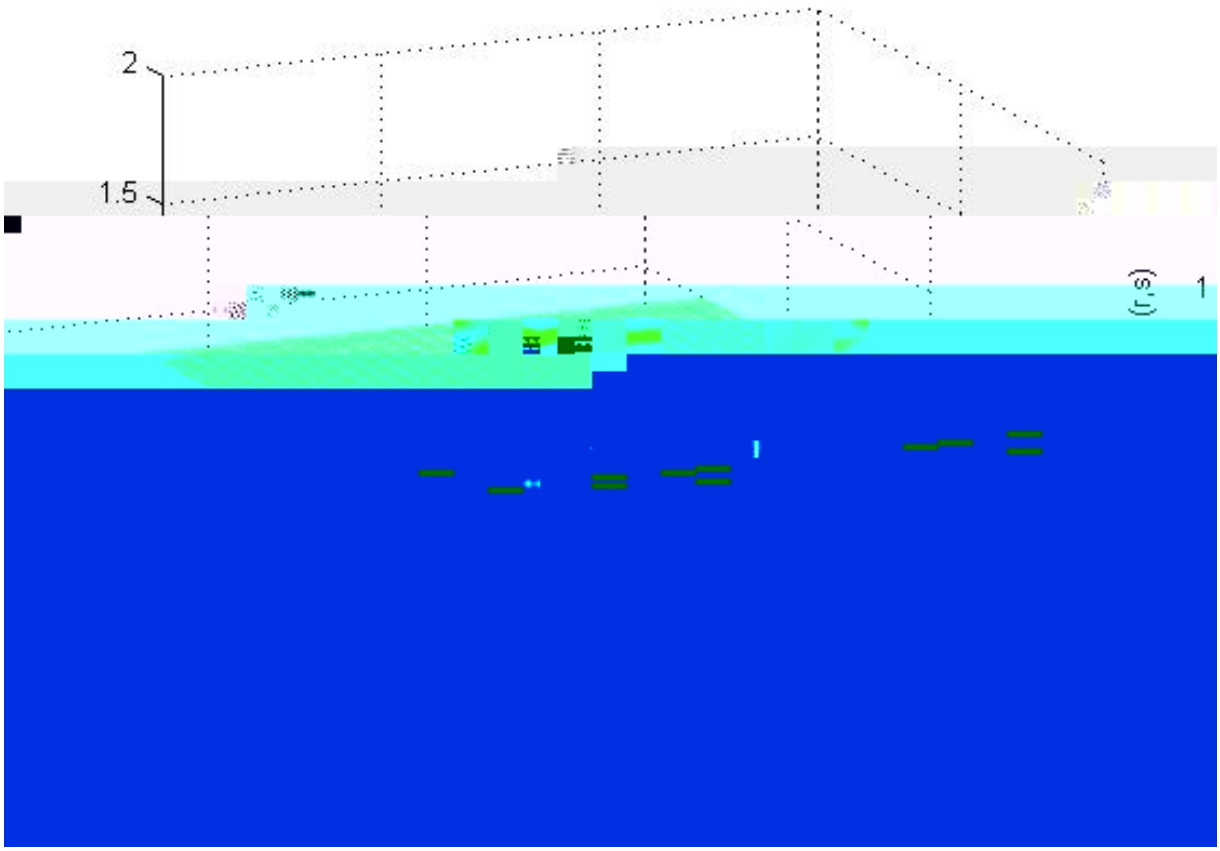


Figure 2.7: q propagator - removing external fields

Now that we have validated that the explicit scheme works correctly, another test run could be to plot the results of the PDE with segragation strength $N = 1$ instead of 100. This would help us understand how the $w_{;i}$ function affects our results and the overall co5at

and from a di erent angle in gure 2.9.

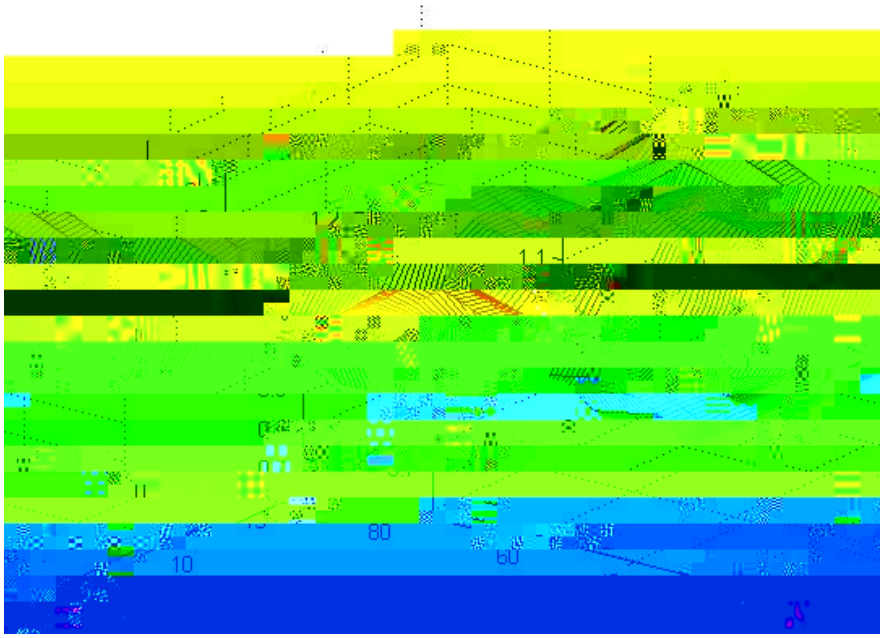


Figure 2.8: q propagator - $N = 1$

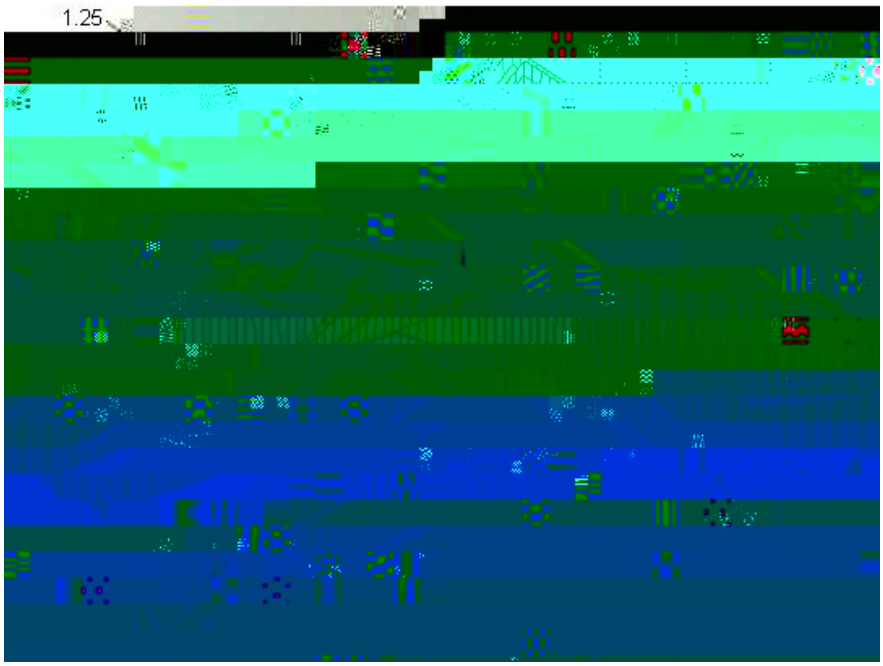


Figure 2.9: q propagator - $N = 1$ - di erent angle

Looking at the results we could see how the $w_{;i}$ function chages the plot from Figure 2.7 to Figure 2.9 and then Figure 2.6. We could therefore conclude that the

larger the values of $w_{;i}$, the larger the values of q will be. An exponential growth appears in the values of q. This could not have been caused by the di usion part of the equation as all the research concludes that di usion satis es the solution which appears in Figure 2.7. We therefore refer to the analytic solution of $\frac{@q}{@s} = w_{;i}q$

remaining di usion equation produces the same result as the q propagator under the same conditions. The discretised equation without the external elds is now,

$$q_{i;\mathbf{j-1}} = q_{i;j} + \frac{1}{6}\frac{ds}{(dr)^2}(q_{i+1;j} \quad 2q_{i;j} + q_{i\ 1;j}) \tag{2.19}$$

and the results satisfy our expectations, as shown in Figure 2.11.

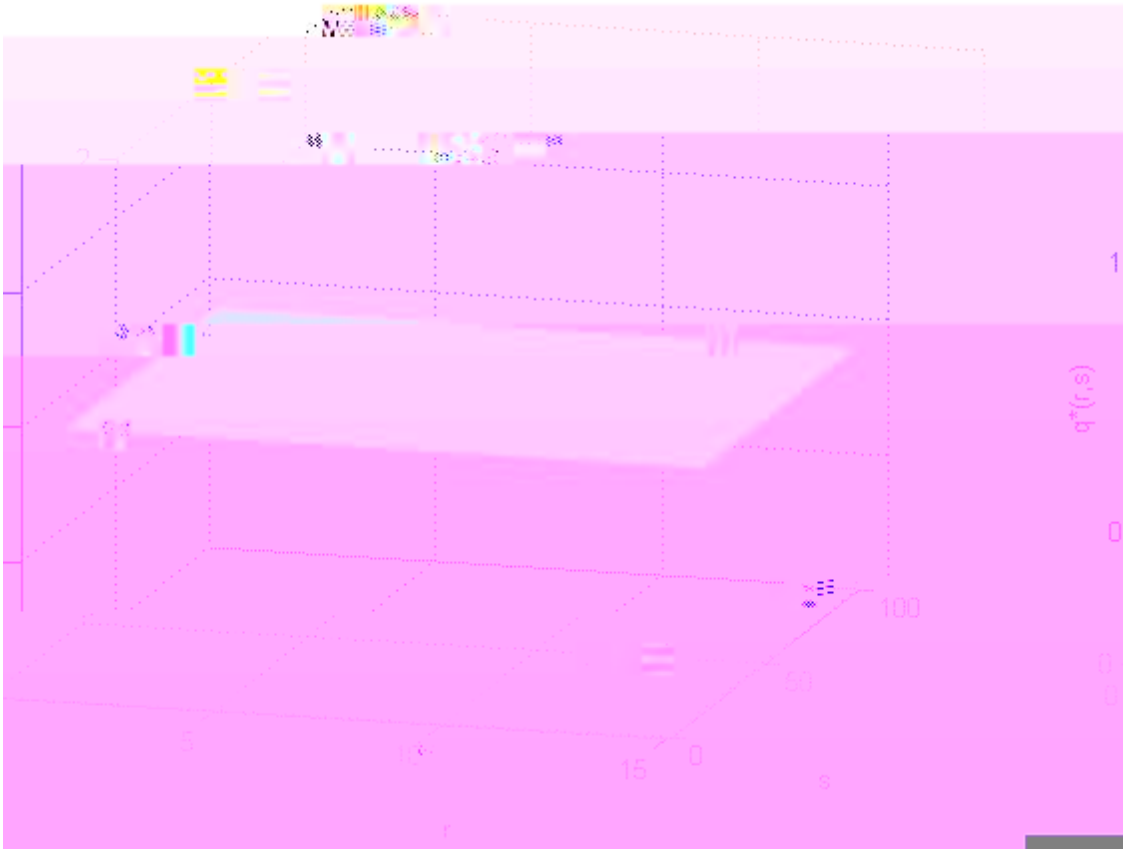


Figure 2.11: q propagator - external elds removed

Plotting the results of the PDE with segregation strength $N = 1$ instead of 100 for q as well, allow us to make better comparisons between the two propagators. That is because the $w_{;i}$ function will be weaker and thus it will have less e ect on the PDE and will not experience such great exponential growth. The outcome is shown in Figure 2.12 and from a di erent angle in gure 2.13.

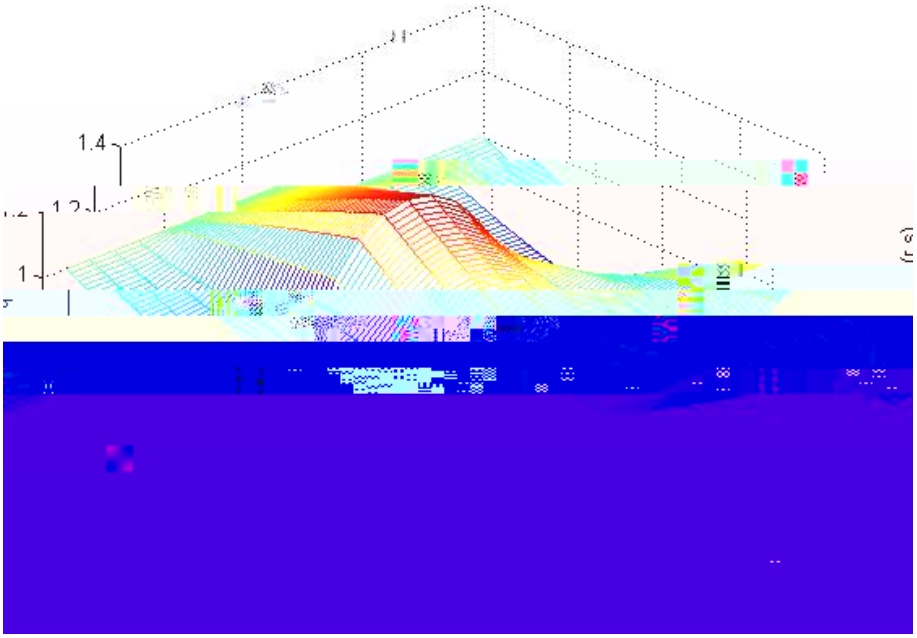


Figure 2.12: q propagator - $N = 1$

Figure 2.13: q propagator - $N = 1$ - different angle

Comparing Figure 2.12 of the q copolymer propagator to Figure 2.8 of copolymer propagator q, we can notice two relatively similar plots, with the first starting from 1 in the last time step, $N_t = 80$, and the second one starting from 1 at the initial time step j=0. They are both relatively symmetrical and q being almost

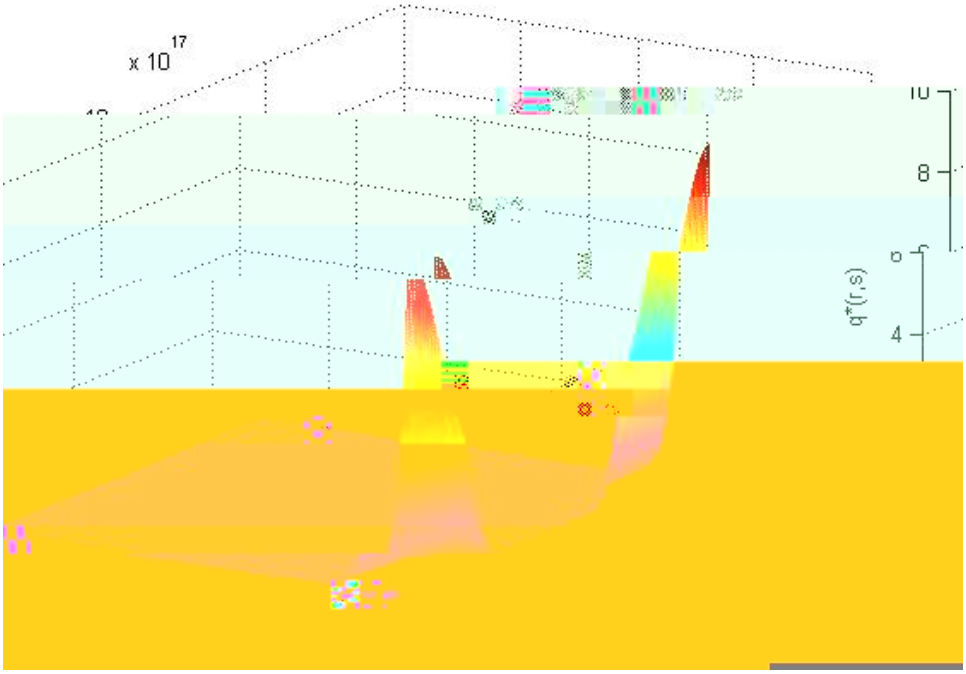


Figure 2.15: q propagator - $N_x = 100$, $N_t = 3592$

2.3 Numerical Integration

Once $q(r;s)$ and $q^{\dagger}(r;s)$ have been calculated using the numerical scheme, we must solve equations 2.20 and 2.21.

$$Q = \int q(r;s)q^{\dagger}(r;s)dr \tag{2.20}$$

$$_A(r) = \frac{V}{Q}\int_0^{Z^f} g(r;s)q^{\dagger}(r;s)ds \qquad _B(r) = \frac{V}{Q}\int_f^{Z^1} q(r;s)q^{\dagger}(r;s)ds \tag{2.21}$$

Equation 2.20 indicates the partition function and 2.21 the ensemble-averaged segment concentration distributions and their results should resemble Figure 1.2b. Both equations involve integrating the product of $q(r;s)$ and $q^{\dagger}(r;s)$,($X = q(r;s):q^{\dagger}(r;s)$). There are several method of numerical integration of varying accuracy and ease

of trapezoids. A trapezoid is a four-sided region with two opposite sides parallel. The area of a trapezoid is the average length of the parallel sides, times the distance between them.
Given the partition [0,D] we can de ne the associated trapezoid sum to correspond to the area under the X-line.

The FORTRAN program will read all the numerical results of

Although the trapezoidal rule is generally only second-order the error is O

approximation of the true result and in fact the numerical solution when plotted in MATLAB is very inaccurate and disapointing, Figure 2.17.

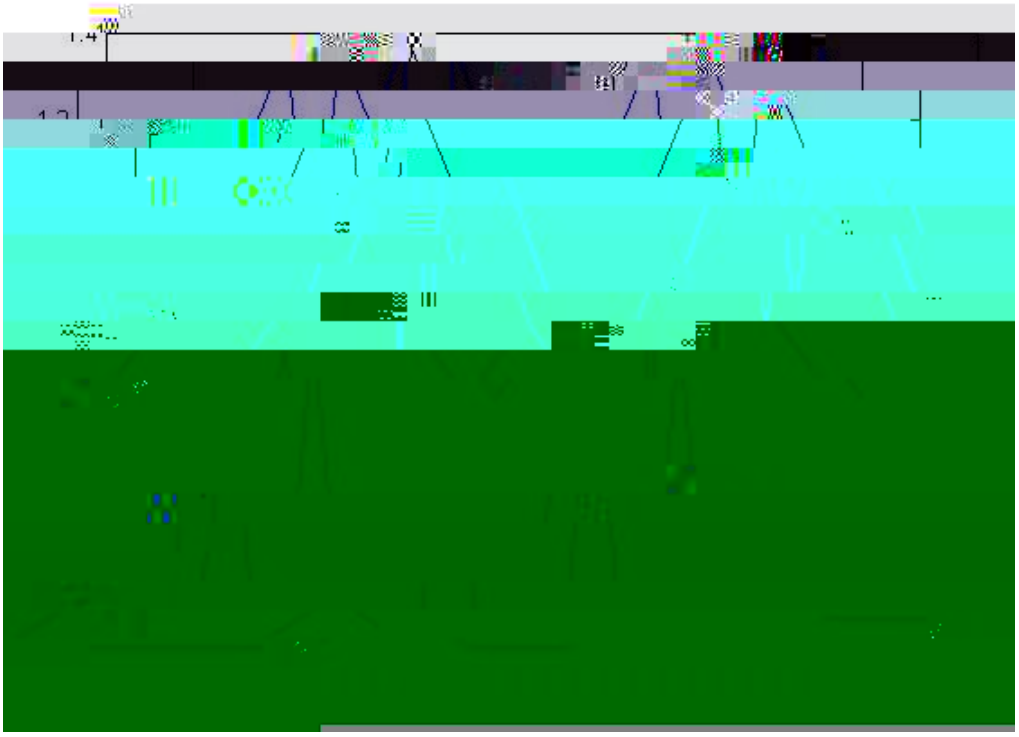


Figure 2.17: Segment Concentration Dist. - N_x = 15, N_t = 80

Another attempt to improve the accuracy and get more realistic results for the segment concentrations (r), is to allocate more space steps, e.g N_x = 100 and corresponding time steps to meet the stability condition, N_t = 3592.

We should point out that an even greater number of space steps is prefered, however by choosing e.g N_x = 1000 we would require N_t = 365533 and therefore we would need a stronger processing unit. This problem is another reason why the explicit scheme is not e cient for our problem.

The results for both Q and $_A$(r), equations 2.22 and 2.23 with N_x = 100 and N_t = 3592 can be observed in Figure 2.18 and 2.19, accordingly. However we can notice that the concentrations are still lacking accuracy and a small error in the last few digits has caused Q not to be a straight line but in fact include a jump of magnitude 113900032 which is still small compare to the values we are handling. The relative di erence is quite small.

All together the explicit scheme is not appropriate for all the reasons mentioned so far, but gives us the background required to use a less unstable scheme and compare

its results.

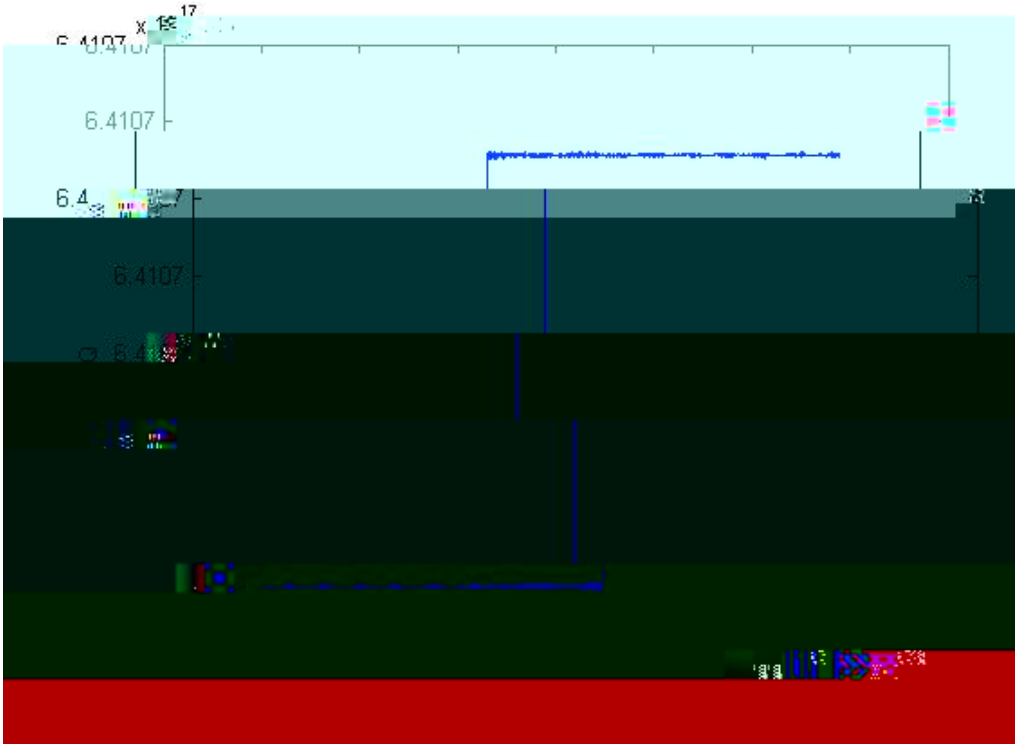


Figure 2.18: Total Partition Function - N_100 = 100, N_t = 3592

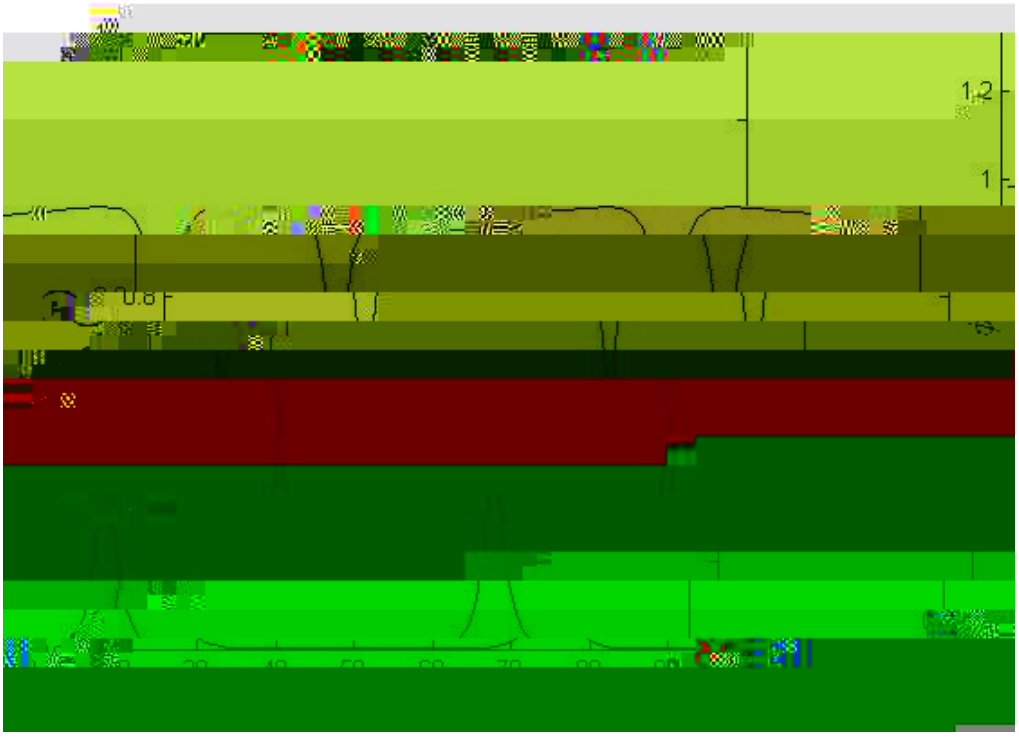


Figure 2.19: Segment Concentration Dist. - N_100 = 100, N_t = 3592

$$\frac{@q}{@s}j_{r_i;s_{j+1}} = \frac{q_{i;j+1} \quad q_{i;j}}{s} + O(\quad s) \tag{3.1}$$

$$\frac{@^2q}{@r^2}j_{r_i} = \frac{q_{i \quad 1;j} \quad 2q_{i;j} + q_{i+1;j}}{r^2} + O(\quad r^2) \tag{3.2}$$

Substituting 3.1 and 3.2 into 2.1 and collecting the truncation errors we obtain

$$\frac{q_{i;j+1} \quad q_{i;j}}{s} = \frac{1}{2}\left(\frac{1}{6}\frac{q_{i \quad 1;j} \quad 2q_{i;j} + q_{i+1;j}}{r^2} + \frac{1}{6}\frac{q_{i \quad 1;j+1} \quad 2q_{i;j+1} + q_{i+1;j+1}}{r^2}\right) + w_{;i}\frac{q_{i;j} + q_{i;j+1}}{2} + O(\quad s^2) + O(\quad r^2)$$

We can notice that the values of q in the above equation from time step j and time step j+1 appear on the right hand side. This equation is used to predict values of q at time j+1 so all values of q at j are assumed to be known. The propagator q is equal to 1 at the rst time step as imposed by the initial data.
Rearranging the above equation so that values of q at time j+1 are on the left (L) and values of q at time j are on the right (R) and dropping the truncation error terms we obtain L = R as follows:

$$L = \quad 1{:}0 + \frac{1}{3} \quad + \frac{s\,w_{;i}}{2} \quad q_{i;j+1} \quad \frac{1}{6} \quad q_{i \quad 1;j+1} \quad \frac{1}{6} \quad q_{i+1;j+1} \tag{3.3}$$

$$R = \quad 1{:}0 \quad \frac{1}{3} \quad \frac{s\,w_{;j}}{2} \quad q_{i;j} + \frac{1}{6} \quad q_{i \quad 1;j} + \frac{1}{6} \quad q_{i+1;j} \tag{3.4}$$

where $= \frac{s}{2 \quad r^2}$.

This equation cannot be rearranged like the explicit equation scheme to obtain a simple algebraic formula for computing for q_i^{j+1} in terms of neighbors like $q_{i+1;j}; q_{i \quad 1;j}$ and $q_{i;j}$.
This equation is one equation in a system of equations for the values of q at the internal nodes of the spatial mesh (i=2,3...,N-1).

The system of equations can be represented in matrix form, the left hand side 3.3 is presented by matrix 3.6. The matrix is tridiagonal and e cient algorithms exist to invert the matrix.
When we perform a von Neumann stability analysis to the scheme by substituting $q_{i;j} = \ ^j e^{ikj \quad r}$ into the di erential scheme, it yields an ampli cation factor:

$$= \frac{1 \quad 2m(sin(\frac{k \quad r}{2}))^2}{1 + 2m(sin(\frac{k \quad r}{2}))^2} \tag{3.5}$$

where $= \frac{k \quad s}{(\quad r)^2}$ and

analysed but with opposite signs. However, since the initial data of q is at the last time step j=N_t, we need to work backwards to time step j=1 and thus we use the same discretised equations with the same signs, for the right hand side, 3.4, 3.7 and 3.8 and for the left hand side the tridiagonal matrix solver for matrix 3.6.

The external field defined by the function $w_{;i}$ and equation 2.16 is computed in the same way as explained in the explicit scheme section 2.2.1 . Once more, for propagator q the first half of the time steps use $w_{A;i}$ and the remaining steps $w_{B;i}$ and for propagator q vice versa. The plotted results should look similar to Figure 2.4.

We will test run the program of the Crank Nicolson scheme with a uniform mesh to compare the results with that of the explicit scheme. Starting with the propagator q and choosing 15 space steps and 80 time steps, the results should look

partial di erential equation. This would transform our PDE into a pure di usion equation of which the discretised form is equation 3.9 and the result should look similar to Figure 2.7 of the explicit scheme.

$q_{i;j}$

to validate that the Crank Nicolson scheme works correctly for the computation of q by removing the $w_{,i}$ function and thus solve an ordinary di usion equation but backwards as the initial data is at the last time step. The solution is 1 throughout the domain and Figure 3.5 presents the results.

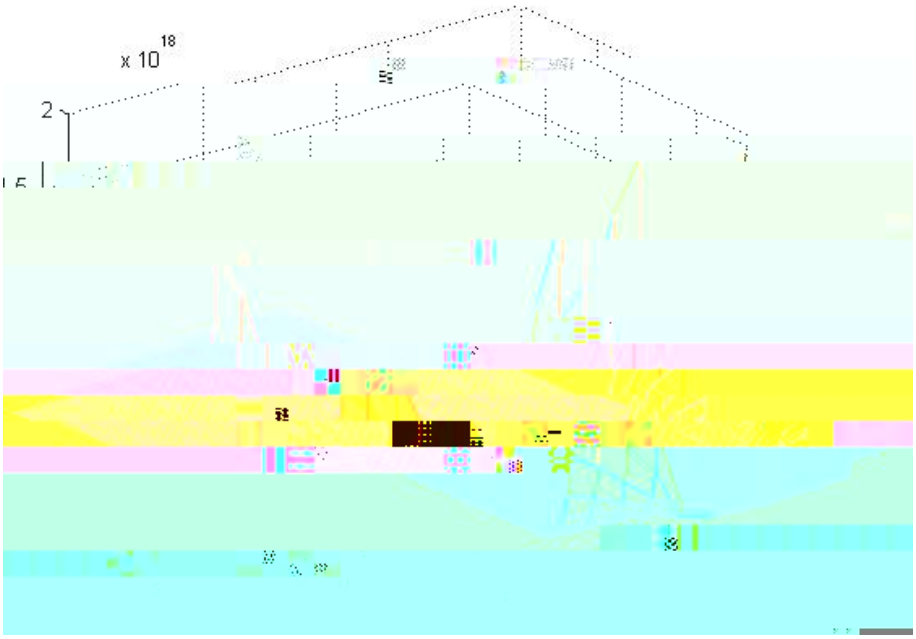


Figure 3.4: q propagator - N_x = 15, N_t = 80

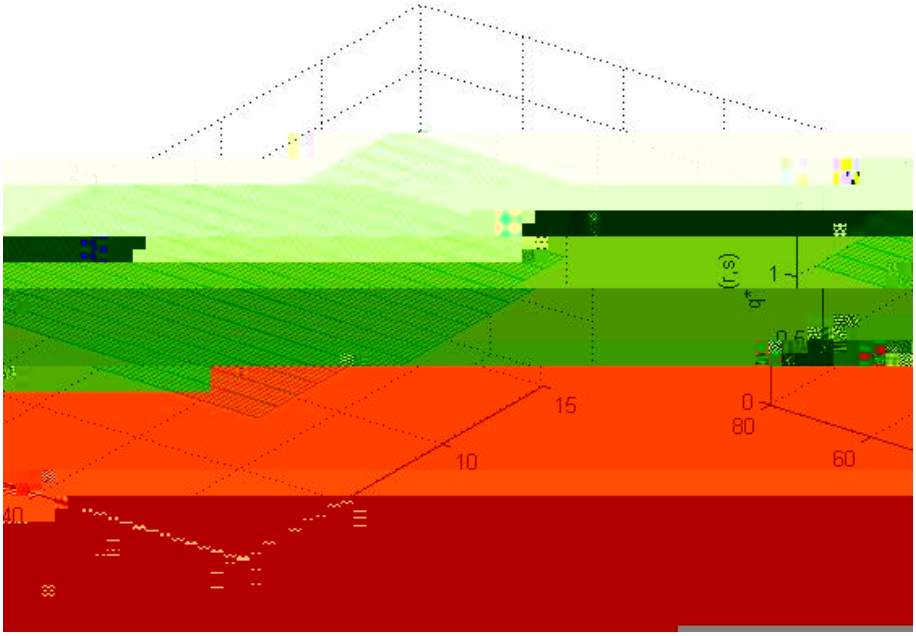


Figure 3.5: q propagator - excluding external elds

Unlike the explicit scheme, there is no stability condition that needs to be satisfied, therefore we can choose a larger number of space steps without having to select a huge number of time steps. We now test run the program on a number of space steps that was difficult to run using the explicit scheme. Figures 3.6 and 3.7 show the results of the q and q propagators for space steps $N_x = 1000$ and time steps $N_t = 100$.

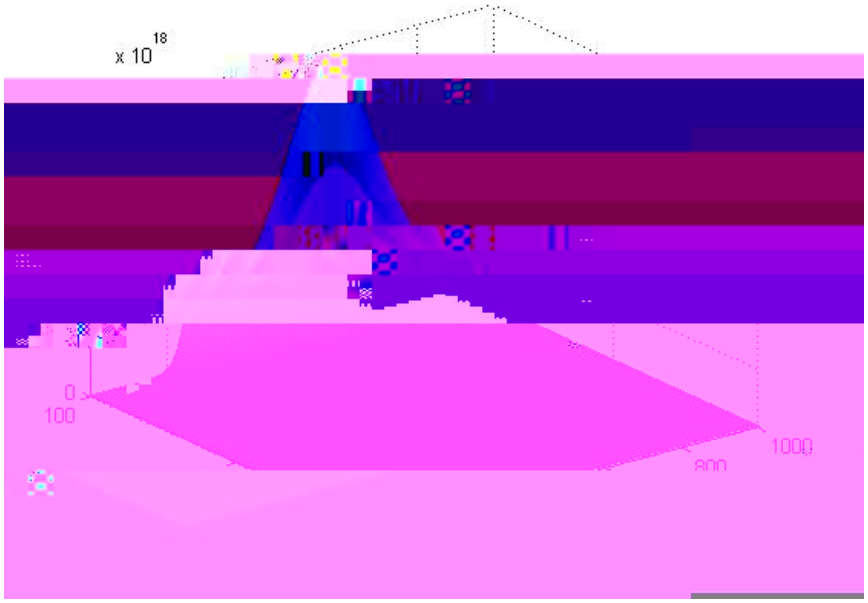


Figure 3.6: q propagator - $N_x = 1000$, $N_t = 100$

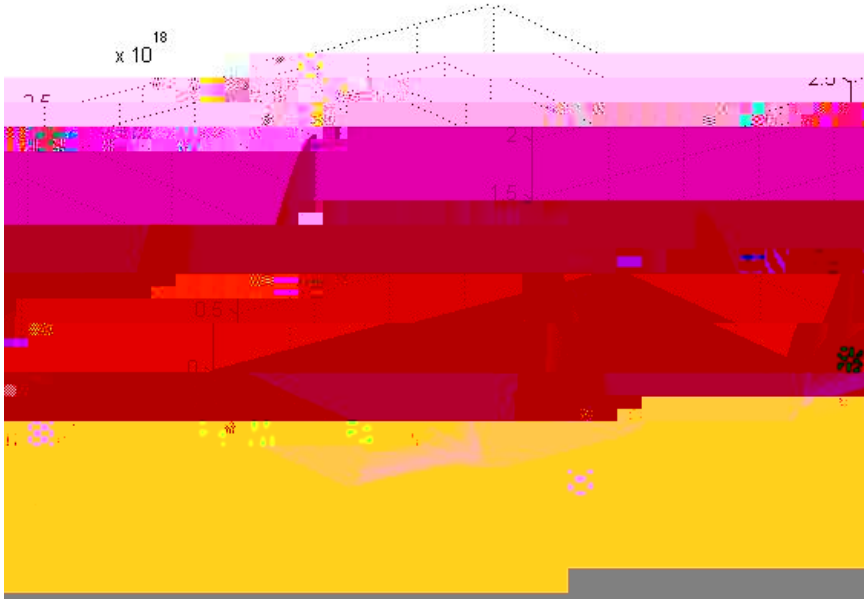


Figure 3.7: q propagator - $N_x = 1000$, $N_t = 100$

We repeat the process and select $N_x = 6000$ and $N_t = 1000$. The results of both q and q propagators are in Figures 3.8 and 3.9. The increase of resolution is noticeable and we only have to refer to the integral equations of Q and (r) to check if the numerical solutions converges to the expected results.

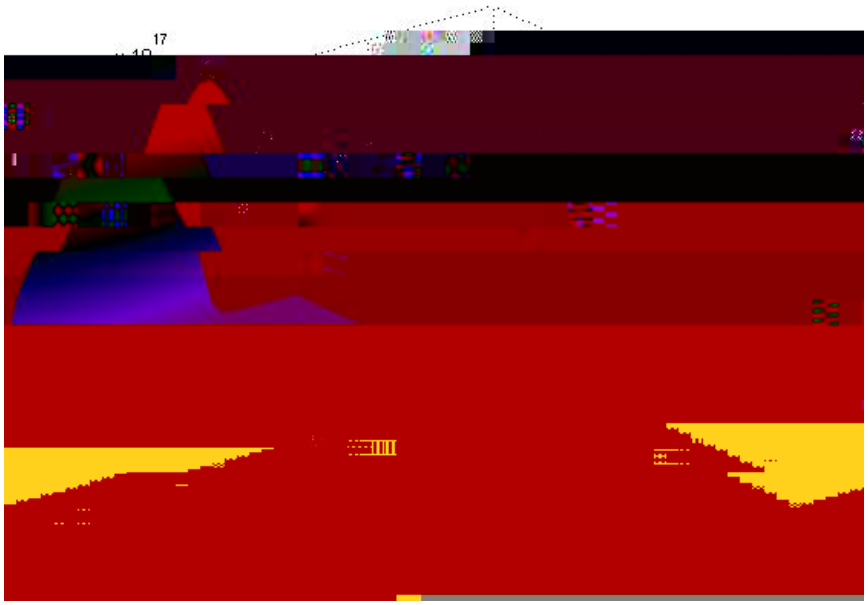


Figure 3.8: q propagator - $N_x = 6000$, $N_t = 1000$

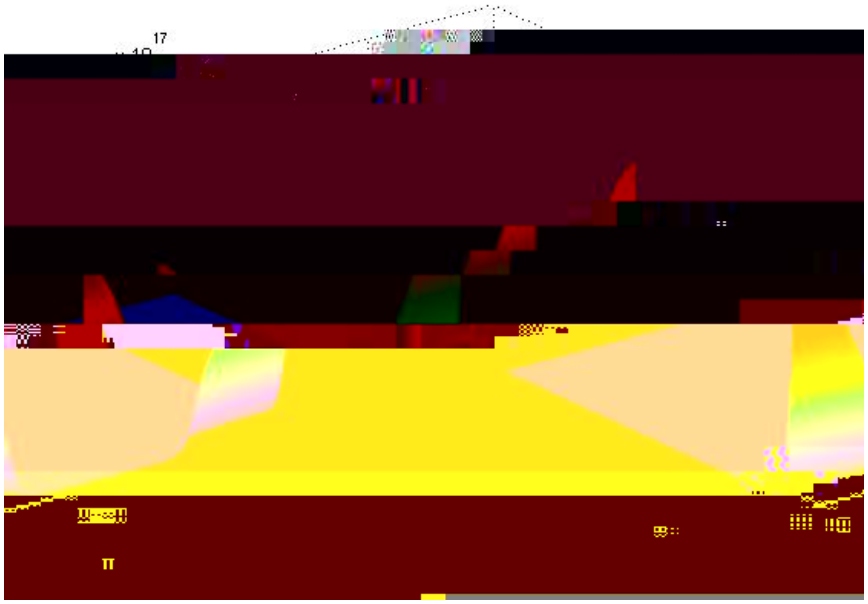


Figure 3.9: q propagator - $N_x = 6000$, $N_t = 1000$

Applying the trapezoidal rule to the integral equations for the total partition

function and the segment concentrations, we should observe similar but more accurate results from the explicit scheme. As we explained at an earlier stage, the numerical solution of Q over all time steps (s points) should be constant and therefore

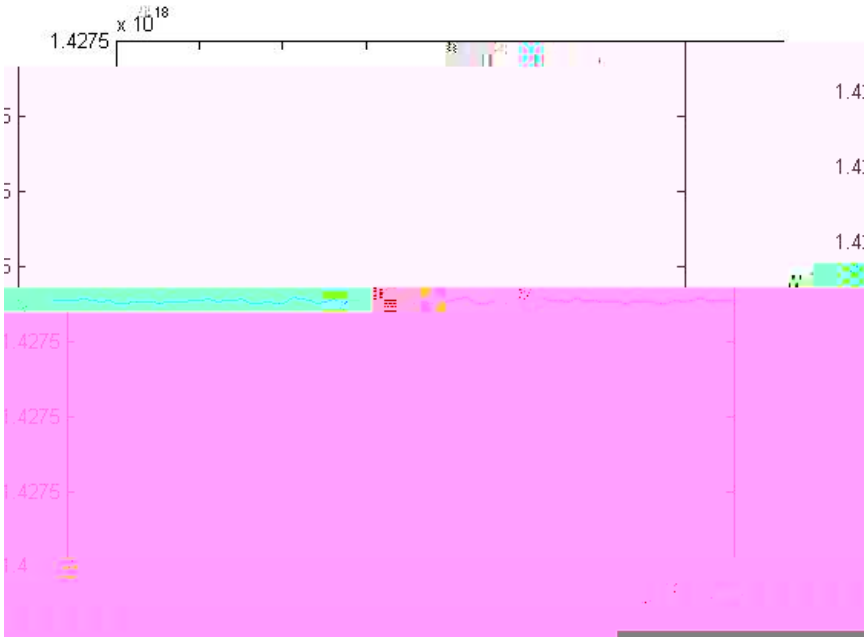


Figure 3.11: Total Partition Function - N_x = 15, N_t = 80

Further on are the results for N_x = 1000 and N_t = 100. We can observe an improvement in the (r

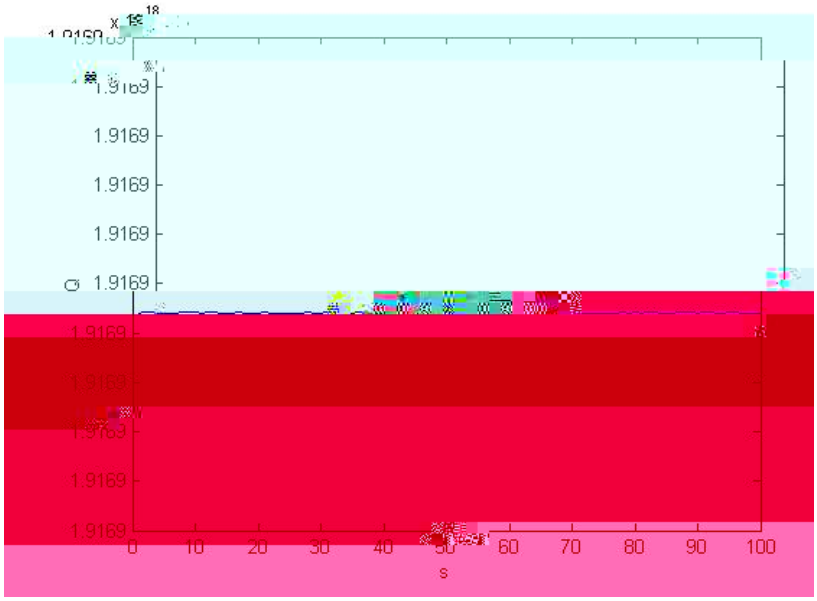


Figure 3.13: Total Partition Function - N_x = 1000, N_t = 100

Finally, we have the results of N_x = 6000 and N_t = 1000. This number of points would be impossible using the explicit scheme with the current processing units available. However as we can see Figure 3.14 resembles the expected results shown in Figure 1.2b and Q is almost a straight line. The magnitude of the greatest oscillation has further decreased to 300032. We must note that the values of Q are very large and this di erence between them is relatively small.

Chapter 4

Grid Re nement

Now that we have created a program that uses the Crank Nicolson scheme and numerically solves our equations on a uniform mesh we would like to increase the level of accuracy at the interface points between the two elds. Having a high level of accuracy at those points is important in order to understand the nature of interactions between the two elds and to manipulate and exploit them in the di erent industrial and commercial applications. In addition, a uniform grid may be disadvantageous when solutions possess large local gradients. In Chapter 1, we have talked about adaptive methods and speci cally about *h re nement*. Using this method we can break the mesh into smaller pieces when necessary and coarsen the mesh where the solution is very smooth if desired [3].

The basic idea of local uniform grid re nement is to cover the spatial domain, D, with nested, ner and ner, locally uniform subgrids so as to accurately resolve steep spatial transitions. This was done to balance the improvement in model accuracy in the area of interest while trying to minimize errors of the re ned model and reduce processing time. When very large re nement ratios are used, errors in the model solution in the coarse section might increase. To avoid that phenomenon, the re nement was done in two stages to achieve a cell size suitable for interactions between the two elds for a single molecule. In addition downsizing to an intermediate scale model made the process more computationally e cient [16].
Stage 1 involves recognising the regions that require re ning and applying the scheme with a new discretised equation and stage 2 aims to smooth the transition between step size as to avoid big "jumps", but to change step size gradually around the areas of interest. A common approach is the h-re nement, to provide re nement in an area of interest. We achieve this by using a nite-di erence grid with variable spacing such that the grid spacing is small where needed and larger away from it.

$$R = 1.0 - \frac{s}{6}\frac{1}{(r_{i+1} - r_i)(r_{i+1} - r_{i-1})} - \frac{s}{6}\frac{1}{(r_i - r_{i-1})(r_{i+1} - r_{i-1})} - \frac{sw_i}{2}\Bigg) q_{i,j} + \frac{s}{6}\frac{1}{(r_{i+1} - r_i)(r_{i+1} - r_{i-1})} q_{i+1,j} + \frac{s}{6}\frac{1}{(r_i - r_{i-1})(r_{i+1} - r_{i-1})} q_{i-1,j} \tag{4.5}$$

As in the previous discretised equation for the Crank Nicolson under uniform mesh, this equation cannot be rearranged to obtain a simple algebraic formula for computing $q_{i,j+1}$ in terms of neighbours like $q_{i+1,j}, q_{i-1,j}$ and $q_{i,j}$. This equation is one equation in a system of equations for the values of q at the internal nodes of the spatial mesh (i=2,3,...N-1).

The system of equations is represented in a matrix form and in fact a tridiagonal matrix, is represented by matrix 4.6,

$$\begin{pmatrix} b_1 & c_1 & 0 & \cdots & 0 \\ a_2 & b_2 & c_2 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & a_{N-1} & b_{N-1} & c_{N-1} \\ 0 & \cdots & 0 & a_N & b_N \end{pmatrix} \tag{4.6}$$

where the coe cients of the interior nodes are:

$$a = -\frac{s}{6}\frac{1}{(r_i - r_{i-1})(r_{i+1} - r_{i-1})}$$

$$b = 1.0 + \frac{s}{6}\frac{1}{(r_{i+1} - r_i)(r_{i+1} - r_{i-1})} + \frac{s}{6}\frac{1}{(r_i - r_{i-1})(r_{i+1} - r_{i-1})} + \frac{sw_{,i}}{2}$$

$$c = -\frac{s}{6}\frac{1}{(r_{i+1} - r_i)(r_{i+1} - r_{i-1})}$$

Due to the re ective boundary conditions,

a_1 is multiplied by2

c_{N_x} is also multiplied by2

Similarly the right hand side as de ned by equation 4.5 satis es the re ecitve boundary conditions by imposing the following equations for the rst and nal space

step, 4.7 and 4.8 respectively.

$$R(1) = 1{:}0 \quad \frac{1}{3} \quad \frac{s\,w_{;1}}{2}\; q_{1;j+1} + \frac{1}{3}\; q_{2;j+1} \tag{4.7}$$

$$R(N_x) = 1{:}0 \quad \frac{1}{3} \quad \frac{s\,w_{;1}}{2}\; q_{1;j+1} + \frac{1}{3}\; q_{N_x\ 1;j+1} \tag{4.8}$$

where $= \frac{k\ \ s}{(\ \ r)^2}$

As for the q propagator the equations are identical but are solved using backward steps, starting from the last space step at $j =$

As we can see from Figure 4.1 the areas of interest are between 200 r 400 and 600 r 800, since those are the regions where the elds intersect. These regions correspond to N_x = 1000 steps, we can therefore generalise them for any N_x steps as follows:

$$\frac{N_x}{5} \quad r \quad \frac{2N_x}{5}$$

$$\frac{3N_x}{5} \quad r \quad \frac{4N_x}{5}$$

Using these inequalities we can specify the areas where we need a ner grid and a coarser grid in the rest of the domain.

The program subdivides the elements in those regions in any even number we choose. As an initial test run we choose to double the points within areas of interest, the space step therefore is divided by a factor of 2, ($\frac{dr}{2}$). Such a grid should be more accurate than a uniform grid calculation. As an example, if we choose N_x = 1000 and N_t = 100, the total points the adaptive Crank Nicolson program will calculate are N_x = 1400 and the results are shown in the feedback table the program produces as shown in Figure 4.2

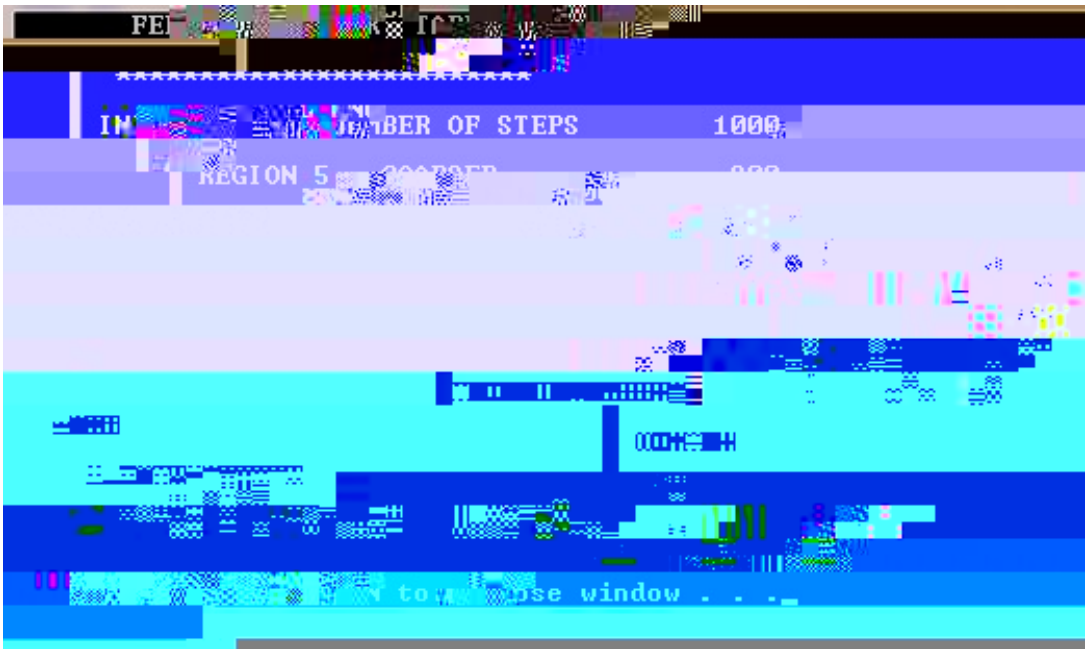


Figure 4.2: Feedback table - Version 6

The propagators q and q that are computed from this program are shown in Figures 4.3 and 4.4, respectively.

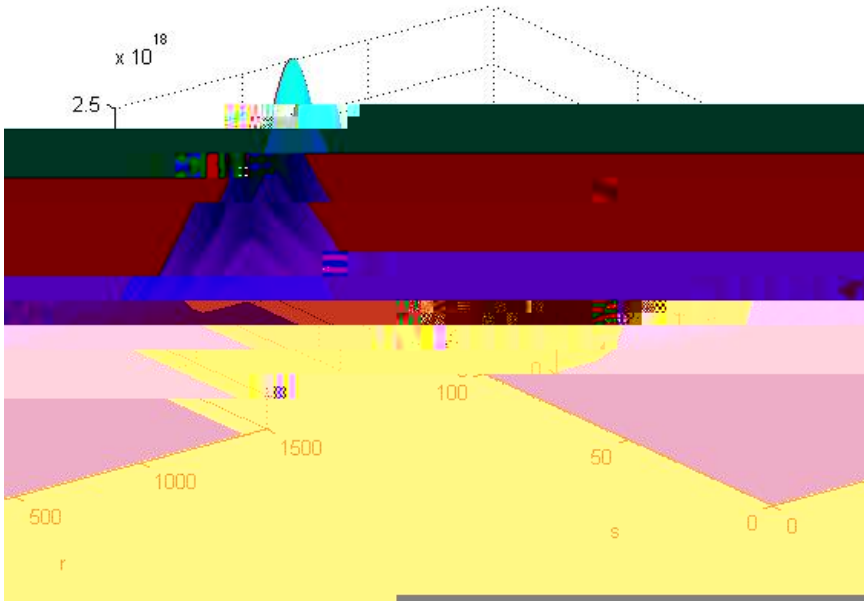


Figure 4.3: q propagator - Version 6 - N_x = 1400, N_t = 100

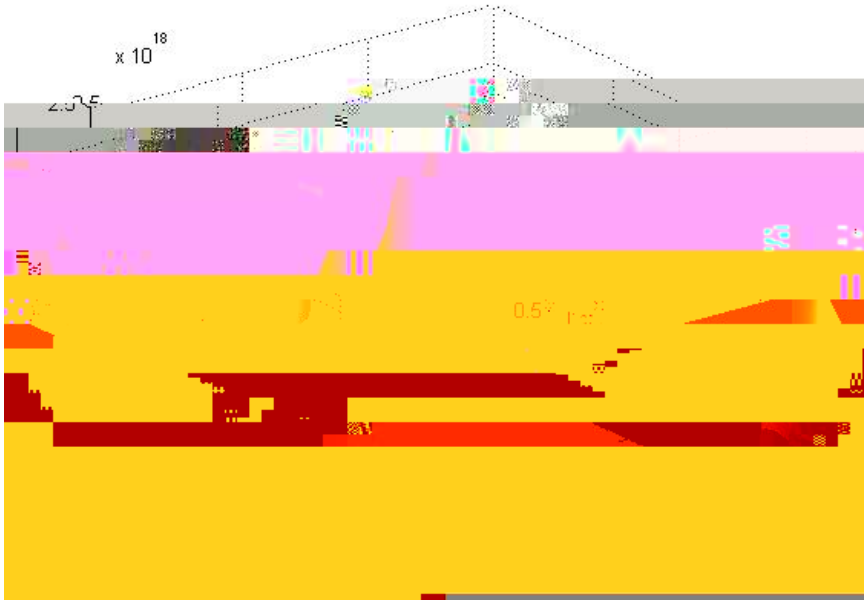


Figure 4.4: q propagator - Version 6 - N_x = 1400, N_t = 100

Finally the trapezoidal rule is used to solve the total partition function Q and the segment concentration distributions (r) are presented in Figures 4.5 and 4.6 accordingly.

a staggering 2.5130e+011. This might be attributed to the sudden change in size steps from coarser to ner regions.

stages. We can select to subdivide the gird by any even factor, obviously the greater the number (k) the ner the grid becomes. As an example if $k=2$ and there are 200 points in the coarse regions 1, 3 and 5, the ne regions 2 and 4 will have 400 points and step size half ($\frac{dr}{2}$) of that of in the coarse regions. The advantages of this adaptive mesh is that less physical memory will be required for a large number of total space steps, N_x, less processing time and power and therefore less costs imposed compared to having a uniform mesh and using the step size $\frac{dr}{2}$ throughout the whole domain. We will explore and compare the time and accuracy di erences between the uniform mesh and di erent versions of the program in more detail in the next Chapter.

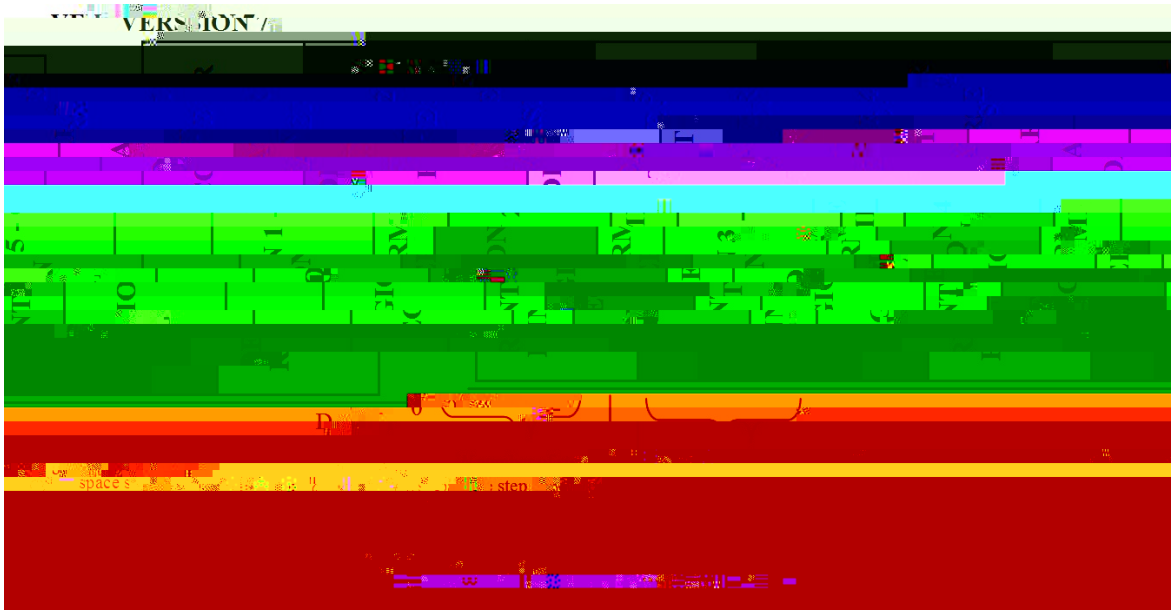


Figure 4.8: Version 7 - Diagram

Program Version 6 is expected to be relatively better than the uniform mesh program, as long as the step size between coarse and ne regions does not di er a lot. The ner the grid in the areas of interest, the greater the accuracy of our results at those points. However, the bigger the "jumps",the greater chances of errors in N.

the step size is $\frac{dr}{}$

The propagators q and q that are computed from this program are shown in Figures 4.10 and 4.11, respectively.

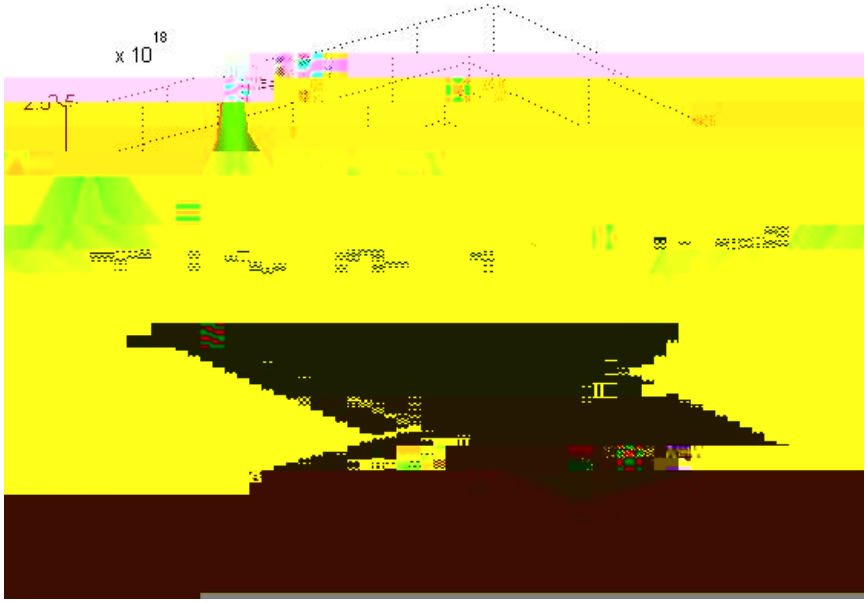


Figure 4.10: q propagator - Version 7

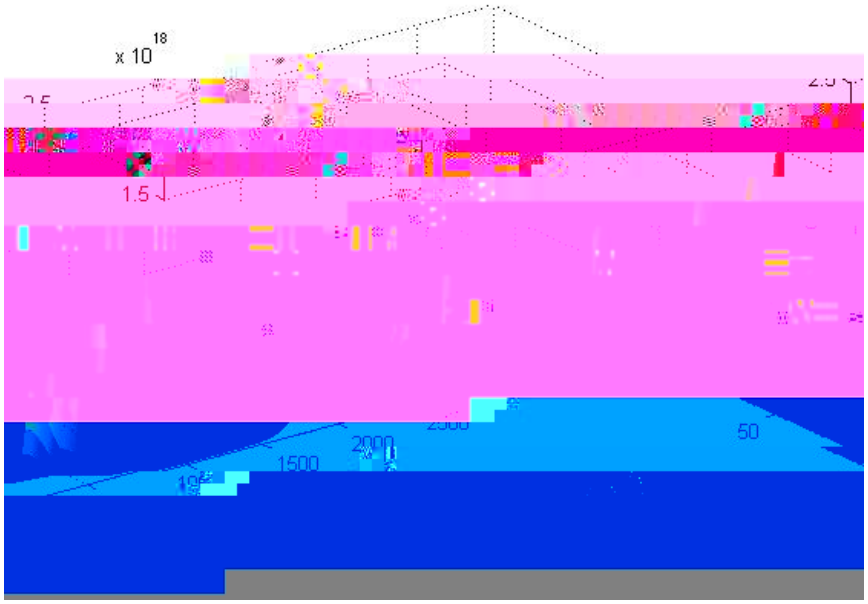


Figure 4.11: q propagator - Version 7

We numerically integrate using the trapezoidal rule to find the total partition function Q and the segment concentration distributions (r). The solutions are presented in Figures 4.12 and 4.13 accordingly.

Chapter 5

Computational E ciency

In order for a local re nement to make sense, re ned calculations need to show some computational savings. Using a ne grid over the entire domain (referred to as global re nement) can be computationally intensive, both in terms of CPU time and memory requirements. We will be investigating which method is more e cient in terms of accuracy, CPU time and memory requirements. The program using a uniform mesh referred to as Version 5, the program Version 6 that uses a locally re ned grid and Version 7 that introduces the concept of a variably spaced grid, were run on the same processing unit, under the same conditions and using a constant total time step, N_t = 100 throughout all the tests. Results are shown for re nement ratios of 2, 4 and 8 in Version 6 and compared to the uniform grid. In each case calculations were performed with a re ned grid and a uniform grid. Version 7 is still under construction and therefore results can only be compared for the re nement ratio of 4. The uniform grid cell size corresponded with the nest cells in the re ned grid. Each calculation was run to the same simulation time and the CPU times were calculated. The CPU times are recorded on the tables below. Our experience has been that the CPU time per iteration does not vary much during a calculation so these timings should be representative.

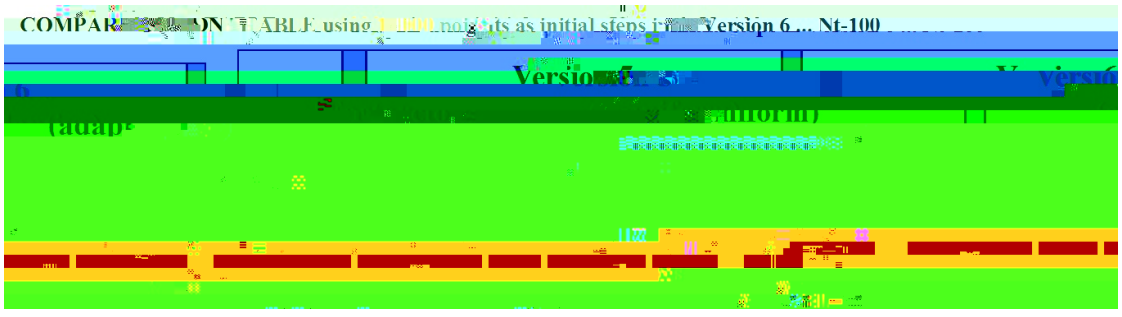
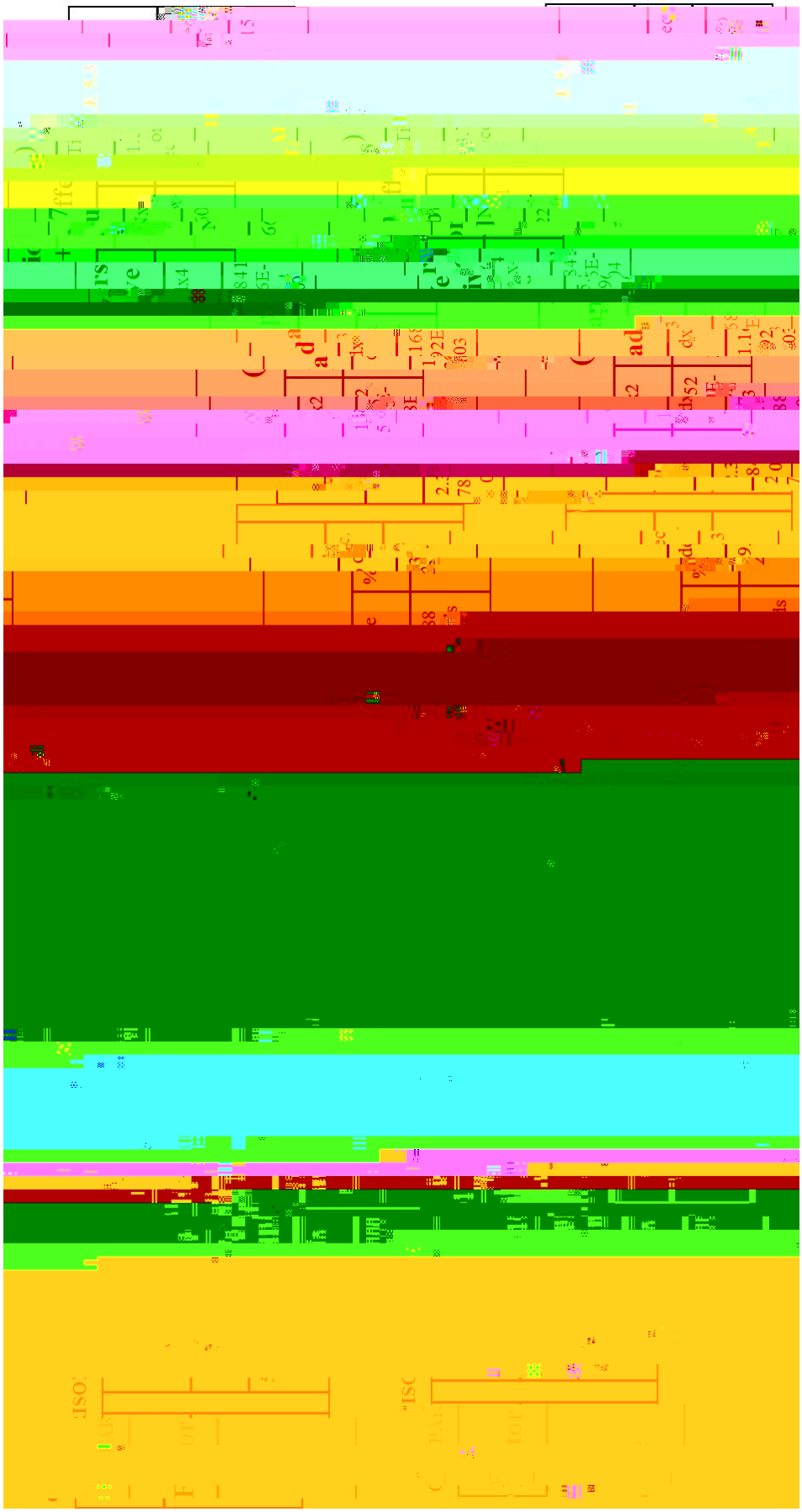


Figure 5.1

As we can see from Table 5.1, by refining the grid at the areas of interest by only a factor of 2, the total CPU time used by the program is decreased by 16.4%. We selected an initial total number of space steps N_x = 1000 in Version 6 of the program. The refinement at the areas around the interface of the external fields were double the points at the rest of the domain and the final points became N_x = 1400. On the other hand by attempting a global refinement using Version 5, points had to be selected as N_x = 2000 so that the size of the space step dr became equal to the step size at the fine regions in Version 6.

Table 5.3 (on the next page) compares all 3 versions of the program. The fine regions of the domain have 4 times the number of points in the coarse regions. However, the difference between Versions 6 and 7 are the extra points. Version 7



Great savings can be made by gradually re ning the grid as to the solution approaches the interface points as in Version 7. Variably spaced grids are still being investigated since we have noticed that the more we re ne the domain covered by the ne cells, the more time the calculations require. Currently the number of cells at the ne regions, in both Version 6 and Version 7, is rather large compared to the coarser levels. A more accurate analysis of the external elds is required so more precise areas of interest are selected.

5.1 Re nement Errors

Applying the local re nement method, by either using a variably spaced grid or by simply splitting the domain in coarser and ner regions, emphasis is given to speci c areas of interest where a ner grid is used. This approach could create errors in calculations of our segment concentration distributions, especially in the parts wher a coarser mesh is used. We therefore want to distinguish the size of the overall error between the results of a globally re ned mesh using the uniform mesh program (Version 5) with small size steps and that of a locally re ned mesh that has similar size steps only at the ner regions.

As an initial test for error calculations we nd the mean sum of $_A(r)$ and $_B(r)$. The incompressibility condition states that $_A(r) + _B(r) = 1$. Using numerical schemes such as the Crank Nicolson and the trapezoidal rule lead us to expect a small variance of the solution to this sum. We therefore calculate the mean sum of the segment concentrations across the whole domain using equation 5.1

$$\text{Mean Sum} = \sum_{r=1}^{N_x} \frac{_A(r) + _B(r)}{N_x} \tag{5.1}$$

The results of this sum are shown in the table below, 5.3. N_t remained constant across all programs and equal to 100. The re nement at the ner regions is done using a factor $k = 4$.

Figure 5.3: Table

From the table above, we can conclude that the more we re ne the grid the greater the error and that can probably be attributed to the coarser areas where less attention is given. However the averages di er only very slightly and the size of the error is not clear and probably not very accurate with this method of measure. We therefore use the error,

$$Error = \sqrt{\frac{1}{D}\int_0^D (\ _{AB1}(r) - \ _{AB2}(r))^2 dr} \tag{5.2}$$

where $_{AB1}(r)$ is the sum of $_A(r) + \ _B(r)$ of the global re nement program which produces the more accurate results throughout the whole domain, since the step size is equal to that of the ner regions in Version 7. $_{AB2}(r)$ is the sum of the segment concentrations as a result of Version 7 that focuses on speci c regions of the domain.

The issue with computing the error using equation 5.2 is that each program makes calculations at di erent points in the domain and the global re nement has a lot more points than the local re nement and we need to use this equation only at the common points. Since we are more interested in the local re nement program, we use its space points as reference where to apply the error equation. However, there might not be an equivalent point in the global re nement program as di erent size steps are used. We therefore use linear interpolation to nd the equivalent $_{AB2}(r)$ at that speci c point.

Once we have a $_{AB2}(r)$ equivalent to $_{AB1}(r)$ at each point that the local re nement program produced, we can then apply equation 5.2. The error equation also includes integrating the di erence over the two totals over the whole domain. We use the trapezoidal rule to numerically solve this integration in the same way as we did for calculating the total partition function and the segment concentration equations.

We will now compare Version 7, a variably spaced grid that re nes the ne re-

gions by a factor of 4 against Version 5, using a globally re ned grid that has equal space steps, to that of the ne regions of the locally re ned grid. We will select N_t = 100 in both programs and the results are shown in Table 5.4

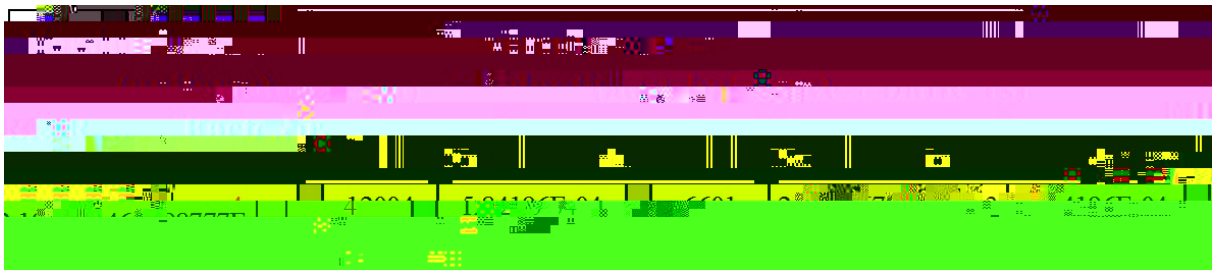
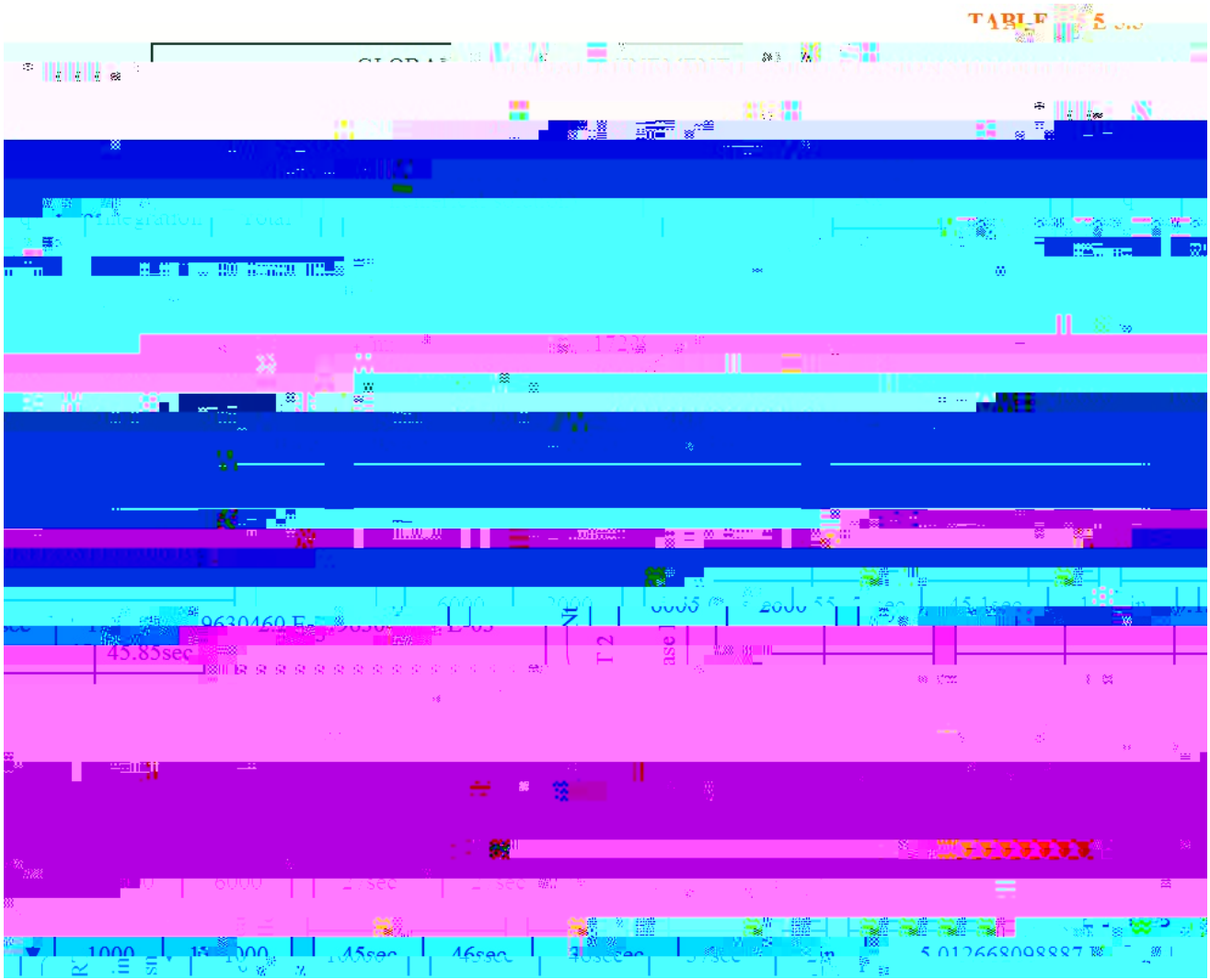


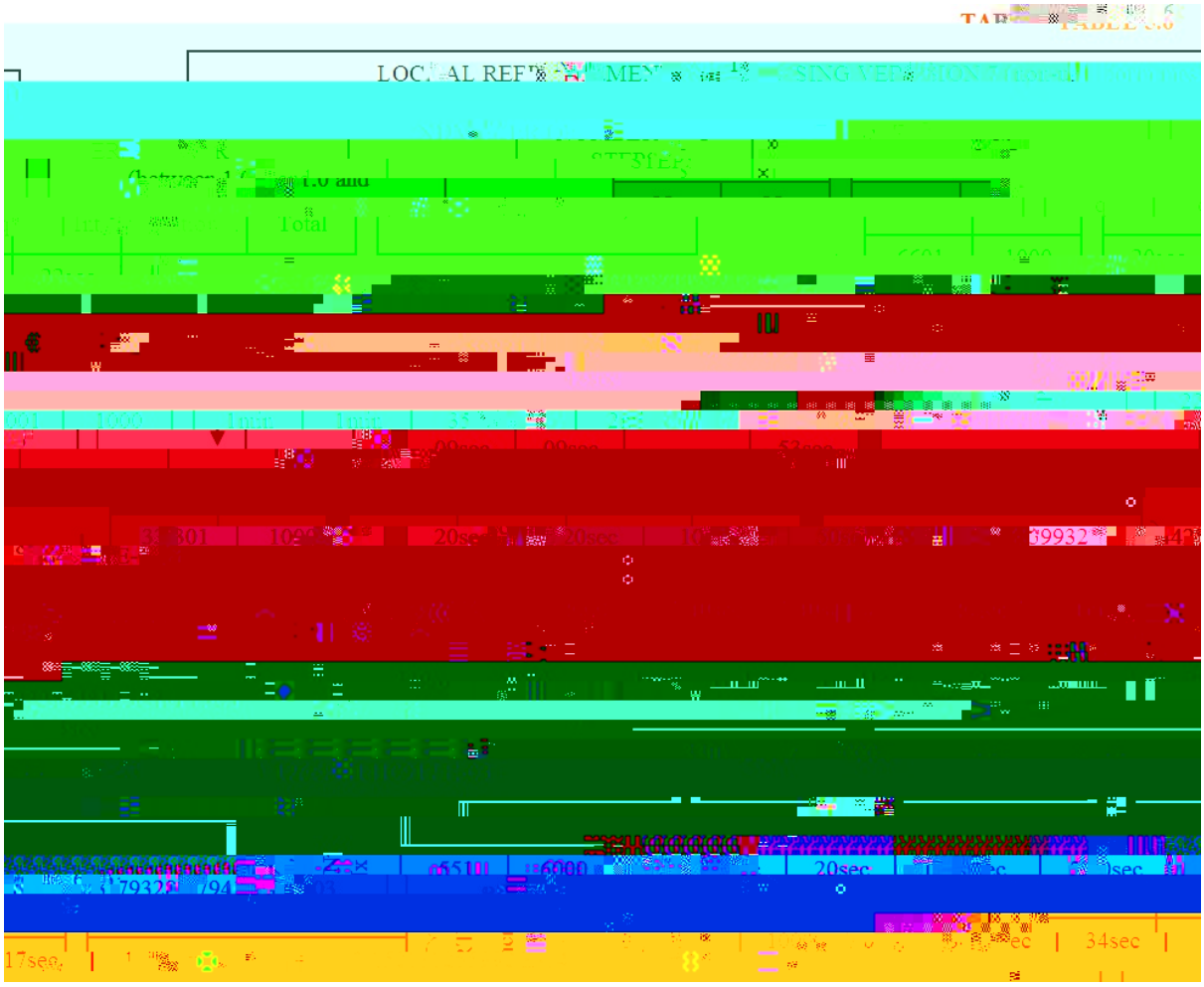
Figure 5.4: error table

The results show that the di erence between the result of globally re ning the grid and using a variably spaced grid is only 2.191146698777E-02. This error is located around the coarser regions of the domain where the locally re ned grid program uses bigger space steps.

Table 5.5 performs global re nement on our domain by using smaller and smaller steps. Part 1 of the table increases the total number of space steps from 12000 to 20000 and then 40000 while maintaining the time steps constant. We can observe a negative correlation between processing time and error. As the processing time increases as a consequence of the extra steps and calculations the error decreases. But it's the change in the error that is surprisingly small. The change appears in the seventh signi cant gure. We therefore perform a di erent test, as shown in Part 2. Space steps are kept at 6000, while we run the program with 1000, 2000, 4000 and 6000 total time steps. The processing time does increase similarly to Part 1, however the error decreases at each stage to almost half the size of the previous test run. Thus having a su cient number of total time steps is important. Part 3 shows a change in total time steps but with a smaller number of space steps. The error increases as expected but very slightly. What is even more surprising is that using 6000 total space steps and 1000 time steps does not a ect the error as much as using 1000 space steps and 6000 time steps. The error is almost double in the rst case. From table 5.5 we can conclude that using 6000 time steps gives ta45 t9We can oy o 6

step size is equal between the two tables at only the ne regions of the domain. The coarser regions in the local re nement process have greater step sizes and are in fact 4 times bigger. We therefore see a big change in the processing time and total space steps required. The error in each test is bigger than the equivalent one using a global re nement but by only a slight variation. The 3 di erent parts lead us to the same observations and conclusions as in Table 5.5. Therefore, the local re nement method reduces the processing time with a cost of a slight increase in the error but maintains all assumptions and conclusions of the more accurate but more demanding in processing power and memory, global re nement approach.





These comparisons show that using a globally re ned grid produces more accurate results, which should be more e cient for analysis and applications. However, the error is very small and is attributed to the coarser regions which are not our areas of interest in the domain. The aim of local re nement is to maintain the accuracy at speci c areas and thus increase the processing speed. In addition the memory requirements are much less severe for re ned grids, provided that the nest levels do not make up a large fraction of the domain. Refering to Table 5.2 the

Bibliography

[1] Hector D. Ceniceros and Glenn H. Fredrickson. Numerical solution of polymer self-consistent field theory. *Multiscale Model. Simul.*, 2(3):452{474, 1993.

[2] Daniel J. Duffy. A critique of the crank nicolson scheme strengths and weaknesses for financial instrument pricing. *WILMOTT magazine*, pages 68{76, 2004.

[3] Joseph E. Flaherty. *Adaptive methods for partial differential equations*. Rensselaer Polytechnic Institute, 1988.

[4] M. Schick G. Gompper. *Soft Matter - Volume 1: Polymer Melts and Mixtures*. Wiley-VCH, 2006.

[5] Ian Gladwell. Numerical integration. *Introduction to Scientific Computing*, 2004.

[6] I. W. Hamle. *Developments in Block Copolymer Science and Technology*. John Wiley and Sons, Lt, 2004.

[7] John Jossey and Anil N. Hirani. Equivalence theorems in numerical analysis:integration, differentiation and integration. 2007.

[8] M W Matsen. The standard gaussian model for block copolymer melts. *Journal of Physics:Condensed Matter*, 14:R21{R47, 2002.

[9] M. W. Matsen and M. Schick. Stable and unstable phases of a diblock copolymer melt. *Physical Review Letters*, 72(16):2660{2663, 1994.

[10] Peter K. Moore. An adaptive h-refinement finite element method for parabolic differential systems in three space dimensions. *SIAM J. Sci. Comput*, 21:1567{1586, 2000.

[11] Gerald W. Recktenwald. Finite-difference approximations to the heat equation recktenwald. *Mechanical engineering (New York, N.Y. 1919)*, 2004.

[12] Mary C. Hill Ste en Mehl and Stanley A. Leake. Comparison of local grid re nement methods for mod ow. *Ground Water*, 44(6):792{796, 2006.

[13] Trueman C.W. Sun, C. Unconditionally stable crank-nicolson scheme for solving two-dimensional maxwell's equations. *IET Electronic papers*, 39(7):595{597, 2003.

[14] Je rey David Vavasour. Self-consistent mean eld theory of the lamellar mor-