Combat Modelling with Partial Differential Equations

By

Therese Keane

A thesis submitted for the degree of

Doctor of Philosophy

School of Mathematics and Statistics

Faculty of Science

The University of New South Wales.
Sydney, Australia.

August 2009

PLEASE TYPE

THE UNIVERSITY OF NEW SOUTH WALES
Thesis/Dissertation Sheet

Surname or Family name: Keane

First name: Therese

Other name/s: Alison

Abbreviation for degree as given in the University calendar: PhD

School: Mathematics and Statistics

Faculty: Science

Title: Combat Modelling with Partial Differential Equations

Abstract 350 words maximum: (PLEASE TYPE)

In Part I of this thesis we extend the Lanchester Ordinary Differential Equations and construct a new physically meaningful set of partial differential equations with the aim of more realistically representing soldier dynamics in order to enable a deeper understanding of the nature of conflict. Spatial force movement and troop interaction components are represented with both local and non-local terms, using techniques developed in biological aggregation modelling. A highly accurate flux limiter numerical method ensuring positivity and mass conservation is used, addressing the difficulties of inadequate methods used in previous research. We are able to reproduce crucial behaviour such as the emergence of cohesive density profiles and troop regrouping after suffering losses in both one and two dimensions which has not been previously achieved in continuous combat modelling.

In Part II, we reproduce for the first time apparently complex cellular automaton behaviour with simple partial differential equations, providing an alternate mechanism through which to analyse this behaviour. Our PDE model easily explains behaviour observed in selected scenarios of the cellular automaton wargame ISAAC without resorting to anthropomorphisation of autonomous 'agents'. The insinuation that agents have a reasoning and planning ability is replaced with a deterministic numerical approximation which encapsulates basic motivational factors and demonstrates a variety of spatial behaviours approximating the mean behaviour of the ISAAC scenarios. All scenarios presented here highlight the dangers associated with attributing \emph{intelligent} reasoning to behaviour shown, when this can be explained quite simply through the effects of the terms in our equations. A continuum of forces is able to behave in a manner similar to a collection of individual autonomous agents, and shows decentralised self-organisation and adaptation of tactics to suit a variety of combat situations. We illustrate the ability of our model to incorporate new tactics through the example of introducing a density tactic, and suggest areas for further research.

Declaration relating to disposition of project thesis/dissertation

I hereby grant to the University of New South Wales or its agents the right to archive and to make available my thesis or dissertation in whole or in part in the University libraries in all forms of media, now or here after known, subject to the provisions of the Copyright Act 1968. I retain all property rights, such as patent rights. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

I also authorise University Microfilms to use the 350 word abstract of my thesis in Dissertation Abstracts International (this is applicable to doctoral theses only).

……………………………………………………………… ……………………………………..……………… ……….…………………..……..…
Signature Witness Date

The University recognises that there may be exceptional circumstances requiring restrictions on copying or conditions on use. Requests for restriction for a period of up to 2 years must be made in writing. Requests for a longer period of restriction may be considered in exceptional circumstances and require the approval of the Dean of Graduate Research.

FOR OFFICE USE ONLY Date of completion of requirements for Award:

THIS SHEET IS TO BE GLUED TO THE INSIDE FRONT COVER OF THE THESIS

Contents

Acknowledgements

By far the greatest thanks must go to my supervisors Professor Jim Franklin and Associate Professor Gary Froyland for their guidance, care, support and patience throughout this thesis and especially during my time in Brisbane. Equal thanks must certainly go to my supervisors and co-workers at the Defence Science and Technology Organisation. In particular Dr Alan Theobald for his support and belief in me, Dr Daniel Conley, Dr Bernard Kachoyan, Dr Jim Smelt and the superb and unequalled amphibious team. I am also very grateful for the support from the University of New South Wales and the ARC Centre of Excellence for Mathematics and Statistics of Complex Systems.

Without the love, support, understanding and unwavering belief of my other half, Adrian, this experience would not have been. Also the unconditional support from my family has been appreciated beyond words.

Therese Keane.

Declaration

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the projects design and conception or in style, presentation and linguistic expression is acknowledged.

Signed ....

Date ....

Copyright Statement

I hereby grant the UNSW or its agents the right to archive and to make available my thesis or dissertation in whole or part in the University libraries in all forms of media, now or here after known, subject to the provisions of the Copyright Act 1968. I retain all proprietary rights, such as patent rights. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

I also authorise University Microfilms to use the three hundred and fifty word abstract of my thesis in Dissertation Abstract International.

I have either used no substantial portions of copyright material in my thesis or I have obtained permission to use copyright material.

Signed ....

Date ....

Authenticity Statement

I certify that the Library deposit digital copy is a direct equivalent of the final officially approved version of my thesis. No emendation of content has occurred and if there are any minor variations in formatting, they are the result of the conversion to digital format.

Signed ....

Date ....

Supporting Publications

Keane, Therese (2009) ‘Combat Modelling with Partial Differential Equations’, submitted to Applied Mathematical Modelling.

Keane, Therese (2009) ‘Partial differential equations versus cellular automata for modeling combat’, submitted to Discrete and Continuous Dynamical Systems - Series B, DCDSB0293.

Abstract

In Part I of this thesis we extend Lanchester's Ordinary Differential Equations and construct a new physically meaningful set of partial differential equations with the aim of more realistically representing soldier dynamics in order to enable a deeper understanding of the nature of conflict. Spatial force movement and troop interaction components are represented with both local and non-local terms, using techniques developed in biological aggregation modelling. A highly accurate flux limiter numerical method ensuring positivity and mass conservation is used, addressing the difficulties of inadequate methods used in previous research. We are able to reproduce crucial behaviour such as the emergence of cohesive density profiles and troop regrouping after suffering losses in both one and two dimensions which has not been previously achieved in continuous combat modelling.

In Part II, we reproduce for the first time apparently complex cellular automaton behaviour with simple partial differential equations, providing an alternate mechanism through which to analyse this behaviour. Our PDE model easily explains behaviour observed in selected scenarios of the cellular automaton wargame ISAAC without resorting to anthropomorphisation of autonomous 'agents'. The insinuation that agents have a reasoning and planning ability is replaced with a deterministic numerical approximation which encapsulates basic motivational factors and demonstrates a variety of spatial behaviours approximating the mean behaviour of the ISAAC scenarios. All scenarios presented here highlight the dangers associated with attributing *intelligent* reasoning to behaviour shown, when this can be explained quite simply through the effects of the terms in our equations. A continuum of forces is able to behave in a manner similar to a collection of individual autonomous agents, and shows decentralised self-organisation and adaptation of tactics to suit a variety of combat situations. We illustrate the ability of our model to incorporate new tactics through the example of introducing a density tactic, and suggest areas for further research.

Introduction

Continuous time approaches to combat modelling have not received a great deal of attention or development since the seminal research of Lanchester [23] in 1914 when he developed the set of ordinary differential equations:

Square Law for Collective combat:

$$\frac{du}{dt} = -k_u v(t), \quad u(0) = u_0, \quad k_u > 0 \tag{0.1}$$

$$\frac{dv}{dt} = -k_v u(t), \quad v(0) = v_0, \quad k_v > 0 \tag{0.2}$$

Linear Law for Individual combat:

$$\frac{du}{dt} = -k_{vu} v(t) u(t), \quad u(0) = u_0, \quad k_{vu} > 0 \tag{0.3}$$

$$\frac{dv}{dt} = -k_{uv} u(t) v(t), \quad v(0) = v_0, \quad k_{uv} > 0 \tag{0.4}$$

Lanchester modelled force dynamics as two forces $u = u(t)$ and $v = v(t)$ with initial sizes u_0 and v_0 respectively. The constants k_u and k_v are known as *Lanchester attrition-rate coefficients* and the addition of the opposing force's density converts the Collective or Square Law equations (0.1) and (0.2) into the Individual or Linear Law equations (0.3) and (0.4). His set of ordinary differential equations has greatly influenced military decision making for many years and permeate military thought and analysis to this day. However, the underlying Command and Control structure implied by these equations is that all individuals have perfect knowledge of the enemy, systematically killing opponents until a winner is determined either by total annihilation or when force numbers reach a predetermined level. That is, despite variations in communications, weapon lethality, terrain effects, location in the domain etc., individual soldiers are deemed to have equal capabilities and affect each other equally. While perhaps applicable to ancient and outdated forms of warfare, this underlying assumption must be addressed if these equations are to be applied to modern warfare that relies heavily on discrepancies of terrain, communications, sensors, weapon and soldier capabilities. Many military researchers recognise this limitation and are seeking to derive more realistic representations to enable a deeper understanding of the nature of conflict, see for example [7, 8, 17].

Progress has been made by Protopopescu *et al.* [45, 46], Rusu [47] and Jaiswal [20], addressing some of the major criticisms of the ODEs and reflecting the development in numerical techniques and computational ability of the 1980s and 1990s. Spatial and temporal variation and local and non-local firing effects were successfully modelled using standard Method of Lines and Finite Difference techniques, and a suite of basic military manoeuvres demonstrated. Frontal attack, turning manoeuvres, envelopment, infiltration [47], reserve deployment and termination decision rules have been implemented. However, troop formation was an *artefact of initial profiles chosen*; that is, a formation initially set as a bivariate Gaussian remained roughly as such as all scenario results published were stopped after a brief period of force interaction. Furthermore, spatially dependent velocity fields resulted in unacceptable numerical losses, restricting the velocity field to a temporally dependent one. More complex velocity fields or other numerical techniques such as Finite Volume Methods have not been investigated. Recent work by Spradlin *et al.* focused on variations of the firing terms while using either stationary forces or simple velocity terms [50].

The rise of agent-based or cellular-automaton models has received much attention in many disciplines, especially defence related research. Models such as Einstein [19], ISAAC [18], Map Aware Non-uniform Automata (MANA) [25] demonstrate a range of behaviour which appears to hint at some form of underlying structure.

Each individual troop is modelled via a rule set relating to quantifiable capabilities such as fire-power, communications and also intangibles such as morale or desire to remain close to friendly forces. These rules encode the nonlinearities necessary for a more realistic description of warfare. These nonlinearities need to be understood in order to develop specialised tactics based on current capability, or enhance the procurement of future capability. We believe that the rapidly developing spatially and temporally discrete approach would be greatly assisted by the corresponding development of the continuous spatial and temporal approach. Indeed, Ilachinski [17], who has been instrumental in the development of ISAAC, stresses the need for research into nonlinear continuous dynamics, exploitation of analogous biological models and phase-space reconstruction techniques. Lauren compares MANA simulation results with fluid dynamic concepts or transition between laminar and turbulent states and maintenance of force profiles to viscosity [24]. If an appropriate suite of equations can be found and analysed, this may eliminate the need for extensive parametric studies and subsequent data mining in order to find those combinations of parameters that produce behaviour of interest in CA or agent-based models. Continuous models can be more transparent in terms of how parameter changes affect outcomes and thus more understandable.

In this thesis, we combine aggregation modelling with traditional Lanchester attrition terms and nonlocal firing terms to more accurately represent the spatial aspects of combat. Initially a one dimensional model is developed and tested to establish the parameters required for the selected numerical technique to be effective. Using simple interaction or firing terms, we compare our results to previous work and demonstrate that cohesive troop movement can be achieved and maintained, even when suffering loss of density through attrition. We extend these equations to two dimensions and again show cohesive troop movement that in previous work had only been achieved artificially by using a carefully chosen initial distribution, low diffusion constant, and a sufficiently short overall simulation time to prevent excessive diffusion. Also, for the first time the inclusion and navigation of obstacles in the domain is possible in a continuous combat model. Again in previous work this had been artificially achieved through the manipulation of the convection vector to form a predefined route.

In the second part of this thesis we compare, for the first time, our set of continuous equations to a series of published scenarios generated using the stochastic cellular automaton wargame ISAAC. This approach provides an alternate mechanism through which to analyse the behaviour seen in the ISAAC scenarios. By doing this, the danger of the reliance on using one instance of ISAAC to derive conclusions is highlighted as our model is shown to demonstrate an equivalent mean behaviour. The tendency for anthropomorphisation of agents when seemingly intelligent behaviour is observed is also shown to be inappropriate as the factors influencing behaviour are more easily understandable in the PDEs. This transparency of our PDE terms also facilitates easy changing or addition of tactics (spatial terms) and/or motivational (social) factors, and the subsequent changes in dynamics more understandable. A series of reverse comparisons is then undertaken to show that the density limitations of the ISAAC construct prevent ISAAC demonstrating the equivalent behaviour as shown in a basic set of PDE results.

Finally we demonstrate how variations in the spatial or social dynamics terms correspond to the employment of different tactics. A density response to enemy detection in the form of either expansion or contraction is introduced and the impact on observed behaviour and density loss throughout the simulation discussed. This highlights the importance of the appropriate choice of tactics in order to successfully represent those aspects of combat in question. Modifying our equations in this manner may also provide inspiration to the combat modelling cellular automata community with regards to further developing agent interactions.

In our model forces and firing coefficients remain homogeneous, a criticism of the traditional Lanchester approach. However the nonlinear nature of the equations is able to mimic those seen in ISAAC that have been labelled as complex. Therefore we show that a continuum of forces is able to behave in a manner similar to a collection of individual autonomous agents, shows decentralised self-organisation, and an adaptation of tactics to suit a variety of combat situations.

Part I

Model Development

In this section we discuss some of the restrictions and shortfalls of the Lanchester Equations, several recent significant extensions of these equations and other developments, especially within the field of mathematical biology, which partly forms the inspiration for this research.

We then develop a new set of combat equations with a physically meaningful representation of the spatial dynamics of soldier movement throughout the battlespace. Interaction terms (weapons effects) retain the usual form as found in the Lanchester equations, in addition to a nonlocal form corresponding to long range weapons.

Due to the nonlinear nature of these equations, a brief overview of commonly used numerical methods is given and the appropriate numerical technique for this situation is outlined. This is implemented firstly in one dimension and compared to published results in order to ascertain acceptable numerical parameters. Firing effects are introduced and density losses compared to results from Protopopescu *et al.* Once a satisfactory one dimensional model is established, a two dimensional implementation is made. Again, numerical parameters are found, firing effects included and density loss comparisons made.

Finally obstacle effects are introduced so that a better approximation to established wargaming tools can be made in the following part.

Chapter 1

Basic Extensions of the Lanchester Equations

Developed during the First World War, the Lanchester Equations have greatly influenced military decision making for many years, still permeating military thought and analysis to this day. Developed in an attempt to elucidate general principles of combat, they do however frequently suffer from inappropriate usage when applied to historical data [43] and general wargaming. Generally the difficulty arises when the equations are simplistically applied to the entire battle and battlefield which may encompass large numbers of opposing forces engaged in multiple distributed skirmishes. That is, the spatial distribution of these forces and whether each force is coordinated as a whole is not considered. The underlying Command and Control structure implied by this type of application is that all individuals have perfect knowledge of the enemy, are able to engage the enemy with equal efficacy regardless of physical separation or decentralised control, systematically killing opponents until a winner is determined. This is exactly how one would expect equations with no spatial element to be interpreted. Despite this apparent simplicity in application of the equations, Lanchester acknowledged that battles should be treated as a series of simultaneous smaller sub battles spatially and temporally separated in the battlespace, with each sub battle analysed separately using these laws. Sub battle results should then be amalgamated to produce an overall result. In this way, the researcher effectively performs a type of pre-processing spatial analysis. However the temptation to treat the entire battle as one large conflict and the difficulty in accurately distinguishing potentially numerous sub-battles remains a considerable difficulty for the analyst. The commonly heard three-to-one force ratio required to ensure victory demonstrates the temptation to reduce the modern complex, complicated and technological battlefield into a simplistic numerical problem. Within the defence community there also remains some confusion in the application of the Square and Linear laws with regards to firing interpretations [22] and the validity of assuming the attrition coefficients are universal [15]. Usually, the Square law is taken to represent aimed or direct fire, and the Linear law representative of area or indirect fire. Depending upon the conditions, both laws may be used to represent both aimed and area fire [43].

Most research using the LEs focuses on either extending the equations in some way in order to account for variations of tactics seen in combat, and/or the application of the original or extended LEs to historical data. An example of one such extension is to use both the Linear and Square laws in representing guerrilla warfare as proposed by Deitchman [5]. Guerrilla warfare is a form of combat where a numerically inferior force with less effective weapons engages a larger force with more effective weapons in a series of skirmishes or small scale engagements. Success of the guerrillas is determined by employing either of two differing methods with the assumption that the larger force has been fragmented into a number of smaller formations. The first method relies on the guerrillas sequentially engaging each small formation while maintaining local numerical superiority during each of these small skirmishes. This method is more in keeping with Lanchester's idea of treating a larger conflict as a series of smaller conflicts. The second assumes partial concealment of the guerrillas in an ambush situation such that the guerrillas have full visibility of their enemy, and thus have the ability to engage all opposing troops simultaneously. For this method the opposing troops are restricted to engaging only this smaller number of visible guerrillas. In Deitchman's paper, intangibles such as moral and psychological effects are discussed although no attempt is made to include them.

Historical data is notoriously limited and when available, may not lend itself to a direct comparison with Lanchester's equations [14, 21], let alone to the spatially dependent combat models as presented here. As we are not concerned with developing a highly realistic set of equations with the purpose of recreating historical battles, comparisons will not be made in this thesis.

As randomness is a significant factor in determining the evolution and outcome of a conflict, many researchers have looked to implement a range of techniques to the LEs to reflect this. Stochasticity, Markov models and fractal attrition models have all been developed however none have proved to be a paradigm shift from the original LEs [43]. Wargames, especially commercial games that include high levels of physical realism, may be appealing due to their detailed reflection of the physical world. However these may generate large data sets requiring sophisticated data mining techniques resulting in great difficulty in extracting or elucidating any underlying general principals.

Any extension of the Lanchester Equations to inherently variable future combat remains an open problem. Using post-conflict data to calculate *a posteriori* knowledge such as the force ratio required for success or informative scenario specific relationships are limited in use. This information is far more valuable before a conflict. The desire for *a priori* analysis producing information divorced from historic particulars which military personnel or analysts can utilise in a general fashion remains the driving factor for military research.

1.1 Existing PDE Models

1.1.1 Protopopescu *et al.*

Protopopescu *et al.* [45, 46] extended the ODE formalism of the original Lanchester Equations to a set of partial differential equations in order to address a major criticism from the military modelling community: the inability to model the movement of forces throughout a domain or battlespace. Exploiting the terrain and enemy positioning within the battlespace by executing the appropriate manoeuvre is a fundamental requirement of all forms of physical warfare both ancient and modern[1]. An example of one such manoeuvre is the classic flanking movement. Here the main body of a force will advance directly towards the enemy, fixing him in position, preventing his retreat and limiting his ability to react to other threats. During this fixing operation conducted by the main force, a subsection will covertly relocate to the flank of the enemy for a surprise attack.

Avoidance of obstacles such as mountains or unnavigable terrain is also a key feature in modern warfare. Modern warfare has moved away from the traditional attritional engagements conducted in large open areas to a form where smaller forces operate in difficult terrain such as the urban environment with greater spatial restrictions. Tactics of urban warfare utilise the built environment to conceal movement, positions and to enable surprise attacks. This has resulted in a new form of warfare, manoeuvre warfare, that relies on the principle of inflicting a disproportionate amount of damage to the enemy's weak points. Spatial dependency is implicit in all of these warfare types and so has been emphasised in Protopopescu's research.

Protopopescu *et al.* consider equations of the following form over the domain $\mathbb{R}^2$. Let $u = u(x, y, t), v = v(x, y, t) : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}$ represent the positive soldier densities of two opposing forces at a given position and time. Let the kernels $k_u(x, y, t), k_v(x, y, t) : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}$ represent non-local interaction between the two forces over some finite domain R.

$$\frac{\partial u}{\partial t} = \nabla \cdot (\mathbf{D_u} \nabla u) + \nabla \cdot (\mathbf{C_u} u) + (a_u + b_u u + k_u * v)\, u + d_u v + e_u \tag{1.1}$$

$$\frac{\partial v}{\partial t} = \nabla \cdot (\mathbf{D_v} \nabla v) + \nabla \cdot (\mathbf{C_v} v) + (a_v + b_v v + k_v * u)\, v + d_v u + e_v \tag{1.2}$$

Where:

$$(k_u * v)(x, t) = \int_R k_u(x - X, t) v_j(X, t)\; dX$$

[1] Psychological or Information operations exploit an enemy's thought and decision process and are not considered in this research.

Taking (1.1) as an example, the physical interpretation of the individual terms is as follows. The first term represents diffusion of the force u. As the soldiers of force u move throughout the battlespace, they will tend to wander or move away slightly with respect to their fellow soldiers, such that the entire force distribution may diffuse over time. As maintaining formation is a critical requisite of a force's overall capability to, for example, respond to threats or move effectively to a specific location, it is assumed that diffusion effects will be minimal. This diffusion may have several dependencies on parameters such as space, time or troop levels,

$$\nabla \cdot (\mathbf{D_u} \nabla u), \quad \mathbf{D_u}(x, y, t) > 0. \tag{1.3}$$

Velocity, or the large scale movement of soldiers toward a predetermined location or in a specific direction, may also have dependencies on several parameters. For example, velocity may represent a desire to move with constant speed to a goal located at the north-east corner of the two dimensional domain. For the examples presented in this paper, the velocity term is set to a constant speed in a fixed direction.

$$\nabla \cdot (\mathbf{C_u} u), \quad \mathbf{C_u}(x, y, t) > 0. \tag{1.4}$$

In this research there are two methods of interaction, aimed and area fire. Aimed fire retains the traditional Lanchester form, representing a close range local interaction between forces comparable to fighting with bayonets or hand-to-hand combat. This is the form most commonly used in predator-prey modelling as colocation of both species is required for predation to occur. In the LEs (eq:LanchesterSquareR) and (eq:LanchesterSquareB) with no spatial element, each soldier is assumed to posses the ability to engage all opposing soldiers simultaneously:

$$d_u v, \quad d_u < 0 \tag{1.5}$$

Area fire now takes on the nonlocal weighted form so that long range weapons can be more realistically represented. Here, the dependency on separation distance of rifle fire or longer range weapons such as artillery or mortar fire is represented by a convolution. The domain of the convolution and the specific form of the kernel k reflect the type of weapon modelled.

$$(k_u * v)\, u. \tag{1.6}$$

In [45] the kernels k_u, k_v were of the form $\beta e^{-\nu_i |x-y|}$ so that the efficacy increases with decreasing distance with a maximum value at zero as expected for say, a rifle. This behaviour is not particularly adequate for modelling artillery fire as there is a minimum distance at which they can operate. We use another form of the above kernel to take this further restriction into account. The remaining terms represent self-suppressing effects due to crowding, environmental constraints etc.,

$$b_u u^2, \quad b_u < 0, \tag{1.7}$$

re-enforcement of the force at a rate dependent upon the number of troops remaining

$$a_u u, \quad a_u > 0, \tag{1.8}$$

or a constant, density independent re-enforcement of the force, present in some forms of the Lanchester equations

$$e_u. \tag{1.9}$$

Setting $a = b = c = e = 0, d \neq 0$ and making appropriate simplifications leads to the modern warfare form of Lanchester's Equations (0.1), (0.2), (0.3) and (0.4). We note that mathematical modelling of factors such as preparation, change of plans or psychological effects such as surprise, motivation, exhaustion or morale and their effects of these on parameters is still to be addressed. While we do not address them in this research, we note that these are useful areas for future research.

The equations (1.1) and (1.2) must be supplemented with initial and boundary conditions. Initial conditions are easily specified and there is great flexibility in defining the initial strength and distribution of each force: Initial conditions:

$$u_i|_{t=0} = u_{i0}(x), i = 1, 2.$$

Protopopescu *et al.* used Gaussian, rectangular, spike and triangular distributions while Jaiswal *et al.* used a bivariate Gaussian distribution. In all cases however, the shape of the initial distribution was not maintained due to diffusion effects. Short simulation times prevented excessive diffusion however this behaviour is not realistic in the military case. Maintaining force formation is essential in modern warfare, a dispersal of soldiers that is not due to actions of the opposing force is unacceptable.

Boundary conditions may be defined in several ways, each of the form:

$$(\alpha_i u_i + \vec{\beta}_i \partial_{\vec{r}} u_i)|_{\vec{r}\in\partial\Omega} = \Upsilon(\vec{r}), i = 1, 2;$$

The general form of boundary conditions for one dimension and force u is:

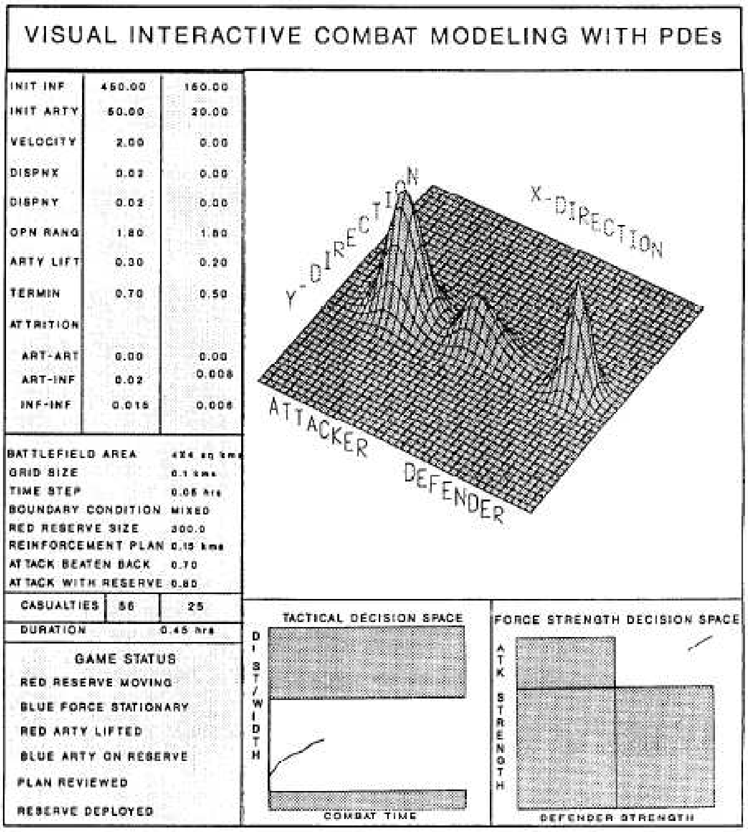
$$a_1 u + b_1 \frac{\partial u}{\partial x} = c_1 \ at\ x = L$$

$$a_2 u + b_2 \frac{\partial u}{\partial x} = c_2 \ at\ x = -L$$

where $a_1, a_2, b_1, b_2, c_1, c_2$ are constants.

For Dirichlet boundary conditions, $a_1, a_2 \neq 0, b_1, b_2 = 0$, Neumann boundary conditions, $a_1, a_2 = 0, b_1, b_2 \neq 0$ and mixed boundary conditions, $a_1, a_2 \neq 0, b_1, b_2 \neq 0$. Mixed boundary conditions were used .

Protopopescu's previous work focused on developing these initial equations and demonstrating that simulation of basic manoeuvres is possible. Various postures or behaviours were imposed on force movement corresponding to predetermined values with each scenario employing varying combinations. These included termination, withdrawal, penetration limits and retreat mechanisms. Withdrawal occurred once a 20% loss of density on either side was attained and was followed by a retreat. A penetration limit halted advancement when 15% of the smaller force penetrated the larger. Termination of the simulation was based on the occurence of a 40% loss by either force. These posture inclusions were seen to produce a higher degree of realism. Jaiswal furthered this approach through the addition of an interface to a suite of termination rules and decision spaces for reserve deployment. This approach taken by both research groups follows a more traditional wargame interface where the operator may run through a suite of simulations, change the decision rules at each iteration, and investigate the differences in outcome - a familiar "what if" analysis tool for the military practitioner. In [20] a simulation may be run in either an Interactive, Partially Interactive or Non-interactive mode. The Non-interactive mode is self explanatory and operates in a similar manner to Protopopescu *et al.* and the other two modes requiring some form of user input during execution of the simulation, involve potential changes to reserve deployment thresholds and termination of combat.



VISUAL INTERACTIVE COMBAT MODELING WITH PDEs

INIT INF	450.00	150.00
INIT ARTY	50.00	20.00
VELOCITY	2.00	0.00
DISPHX	0.02	0.00
DISPHY	0.02	0.00
OPN RANG	1.80	1.80
ARTY LIFT	0.30	0.20
TERMIN	0.70	0.50
ATTRITION		
ART-ART	0.00	0.00
ART-INF	0.02	0.008
INF-INF	0.015	0.006

BATTLEFIELD AREA	4X4 sq kms
GRID SIZE	0.1 kms
TIME STEP	0.05 hrs
BOUNDARY CONDITION	MIXED
RED RESERVE SIZE	300.0
REINFORCEMENT PLAN	0.15 kms
ATTACK BEATEN BACK	0.70
ATTACK WITH RESERVE	0.80

CASUALTIES	56	25
DURATION		0.45 hrs

GAME STATUS

RED RESERVE MOVING

BLUE FORCE STATIONARY

RED ARTY LIFTED

BLUE ARTY ON RESERVE

PLAN REVIEWED

RESERVE DEPLOYED

Figure 1.1: Interface developed by Jaiswal *et al.* Reprinted from [20], with permission from Elsevier.

No further analysis of these approaches has been made, nor comparisons to historical data or other wargaming/conflict models.

1.1.2 Spradlin

Recent work by Spradlin *et al.* [50] focused mainly on investigation of different force interaction terms. The spatial criticisms highlighted in the previous section were addressed in a similar although more simplistic manner than Protopopescu *et al.* Spradlin *et al.*'s two dimensional model used two opposing forces r and b undergoing advection throughout the domain without diffusion and interacting via four variations of the attrition rates $I_r(x, y, t, r, b)$ and $I_b(x, y, t, r, b)$ - Local Direct Fire, Nonlocal Area Fire, Nonlocal Direct Fire: Front Model and Surrounding Direct Fire: Front Model. Forces either remained stationary or simple velocities $v(x, y, t)$ with only one spatial dependency such as $\mathbf{v}(x, y) = (1-x, 0)$ and $\mathbf{v}(x, y) = (3(0.6 - x), 0)$ used.

$$\frac{\partial r}{\partial t} = -\nabla r \cdot \mathbf{v}_r - I_r, \quad \frac{\partial b}{\partial t} = -\nabla b \cdot \mathbf{v}_b - I_b$$

Local Direct Fire:

$$I_r(x, y, t) = \alpha b(x, y, t), \ \alpha > 0$$

Nonlocal Area Fire:

$$I_b(x, y, t) = \left(\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \varphi_r(x - x', y - y') b(x', y', t) dx' dy' \right) b(x, y, t)$$

where $\varphi_r(x, y)$ is chosen to be a bivariate Gaussian. Nonlocal Direct Fire: Front Model:

$$I_b = \int \int_{\mathbb{R}^2} I_{r,b}(x', y', x, 0) dx' dy'$$

where $I_{r,b}$ is the attrition rate of blue forces at (x', y') caused by red forces at (x, y) and is defined as:

$$I_{r,b}(x, y, x', y', t) = A_r r(x, y, t) \frac{\varphi(x' - x, y' - y) b(x', y', t)}{V_{r,b}(x, y)}$$

Surrounding Direct Fire: Front Model where there is only a dependancy on x:

$$I_r = A_b r(x) \left(\int \int_{\mathbb{R}^2} b(x')^{\frac{1}{p}} \frac{\varphi_b(x - x', y')^{\frac{1}{p}}}{V_{b,r}(x')^{\frac{1}{p}}} dy' dx' \right)$$

and similarly for I_b

As with Protopopescu, the desired force profile of constant density interior with sharp edges is an artefact of the initial conditions and short durations of the simulations. *Results from all four firing terms investigated showed not only the expected density loss due to firing terms but a lack of reformation of the forces. There is no reconstitution of the desired force density profile undertaken after losses were incurred. While it is useful to investigate differing forms of the interaction terms, we believe realistic movement throughout the battle space must also be addressed.*

1.2 Mathematical Biology

Lanchester Equations have been used in mathematical biology to provide a framework for the study of animal interactions. Conversely, mathematical biology is a logical research area from which to derive insight and inspiration into other approaches to combat modelling as a significant volume of research has been undertaken in order to understand animal dynamics. In essence, combat can be viewed as an extension of traditional predator-prey interactions. Both parties act as predator and prey however interactions can occur over both local and non-local spatial scales. Birth rates are superseded by resupply rates that do not necessarily depend on current soldier numbers unlike birth rates, and time scales may be far shorter than expected life spans of the participants. For many species there may be a correlation between mortality rates, group size and fighting ability and this may be applicable to combat modelling. Unlike weapons, fighting ability in animals can also be linked to their ability to defend themselves. As will be shown in the following, the similarities between these approaches have been recognised for quite some time.

While we note there are many areas of similarity, there are other aspects of biological modelling that have yet to be incorporated into combat modelling. Swarming, schooling, clumping and self-organisational behaviour in nature is well known and widespread (see [12, 38, 39, 40, 41, 42]) and provides classic examples of complexity despite no obvious external drivers orchestrating the behaviour. Instead, internal forces acting over short ranges are responsible for the large scale coordinated behaviours we observe on a daily basis. Strong social interactions such as attraction, repulsion and collision avoidance are generally effective over larger distances, inferring the need for non-local representation of these effects. These concepts are highly applicable to the military domain as a typical feature of combatant movement is the maintenance of a constant interior density with sharply defined edges.

For continuous reaction-diffusion-advection type models these non-local influences have been incorporated into the advection term in the form of convolutions (or their discrete form for discrete modelling) and represent a balance of attractive and repulsive forces of individuals towards each other [35]. Adopting this approach, cohesive stable swarms with realistic interior densities can be achieved with Mogilner *et al.* determining travelling solutions and Topaz *et al.* [52] providing stationary solutions. Other methods of representing clustering such as the population density dependent acceleration method as described by Tyutyunov *et al.* [55] are also able to reproduce clustering behaviour and are discussed below.

1.2.1 Predator-Prey modelling

Similarities of these predator-prey biological systems to combat have long been recognised by the defence research community. Epstein, an advocate of pursuing a mathematical biology approach to warfare modelling, has demonstrated the derivation of the LEs and the Richardson Arms Race model from both Lotka-Volterra equations and basic biological modelling techniques. Alterations of the Lanchester Equations to include reinforcements and carrying capacities have also been shown to give Gause's model of competition between two species.

1.2.1.1 Lotka-Volterra Equations

Proposed between the First and Second World Wars and so of a similar age to the Lanchester Equations, the Lotka-Volterra equations were derived independently by Lotka in 1925 and Volterra in 1926. Classically this model describes an unrestrained prey species whose population is only restricted through predation, and a predator subject to a given death rate. Further generalisations can produce mutualism or competitive behaviour.

For the prey species x_1 and predator species x_2 with constants $\alpha, \beta, \gamma, \delta$ defining the their interaction, the non-spatial deterministic Lotka-Volterra Equations are:

$$\frac{dx_1}{dt} = x_1 \left(\alpha - \beta x_2\right),$$

$$\frac{dx_2}{dt} = -x_2\left(\gamma - \delta x_1\right),$$

When there are two species competing for a common resource:

$$\frac{dx_1}{dt} = r_1 x_1 \left(1 - \frac{x_1}{K_1} + a_{12}\frac{x_2}{K_1}\right),$$

$$\frac{dx_2}{dt} = r_2 x_2 \left(1 - \frac{x_2}{K_2} + a_{21}\frac{x_1}{K_2}\right).$$

The *intraspecific* competition/environmental limitations on both populations are represented by the constants K_1 and K_2 arising from the classic logistics population model. Each population in isolation will reach equilibrium densities of K_1 and K_2 respectively. a_{12}, a_{21} are the *interspecific competition* or predation rates and usually assumed to be greater than zero.

Three states are possible depending on the values of K_i and a_{ij}.

1. Coexistance: $K_1 > a_{12}K_2$ and $K_2 > a_{21}K_1$ give a stable equilibrium with both populations having a nonzero population.
2. Competitor Exclusion: for example $K_1 < a_{12}K_2$ and $K_2 > a_{21}K_1$ where population 1 predates population 2 to extinction.
3. Founder Control: $K_1 < a_{12}K_2$ and $K_2 < a_{21}K_1$ where initial densities determine the successful population as coexistence is not possible.

Setting $\frac{dx_1}{dt} = \frac{dx_2}{dt} = 0$ in (1.2.1.1) yields the interior equilibrium conditions that are also the equilibrium conditions for the linear system, giving the Richardson Arms Race model:

$$\frac{dx_1}{dt} = r_1 - a_{11}x_1 + a_{12}x_2,$$

$$\frac{dx_2}{dt} = r_2 - a_{21}x_1 + a_{22}x_2,$$

Here a_{11}, a_{22} represent action/reaction, a_{12}, a_{21} fatigue on the economic resources of the country and r_1, r_2 the grievance terms each country has for one another and are assumed to be negative. This model can be further manipulated in order to give the simple epidemic model or SIR (Susceptible, Infected, Recovered). Setting $a_{12} = a_{21} > 0; r_1 = r_2 = a_{11} = a_{22}$, and including recovery rate of infectives proportional to infected population, gives the simple epidemic model. Note that r_2 is now assumed to be positive as it represents the recovery rate.

$$\frac{dx_1}{dt} = -a_{12}x_1x_2,$$

$$\frac{dx_2}{dt} = a_{12}x_1x_2 - r_2x_2$$

Epstein [8] likens the behaviour of this SIR model with the permeation of revolutionary ideals and the tactics employed by a totalitarian government to quash the“infection” of new individuals with these ideals. He does not elaborate further on these simple equations any more than highlighting the fact that many seemingly dissimilar phenomena are described on a gross level by simple related equations with origins in mathematical biology. He uses this to champion a biological approach to modelling human behaviour in the social sciences, particularly warfare.

In his second paper from the 1992 Lectures in Complex Systems [7], his aim was to develop a model of warfare that will produce the core system behaviour. He again gives a brief review on how developing Lanchester Equations through the introduction of, for example, reinforcements or density restrictions gives Gause's model of competition between two species. Gause's principle states that competing species cannot coexist in the same location such that one or both species will eventually become extinct.

Epstein's main criticism of the LEs has been the omission of the spatial considerations of force movement. He addressed this by modelling the velocity of the soldiers on the front line dependent on the force ratio. It was found that the attrition rates remain unaffected whether a withdrawal takes place or not. Epstein argues that although movement is influenced by attrition, the reverse is not the case and goes on to develop the Adaptive Dynamic Model of Combat.

1.2.2 Adaptive Dynamic Model of Combat

This is a system of delay equations based on a generalisation of the LEs where withdrawal now lowers attrition rates. That is, flight mechanism prosecution rates and attrition rates are linked. His key parameters are"equilibrium" attrition rates α_{dT} (defender) and α_{aT} (attacker), and the maximum daily attrition rate the defender/attacker is willing to suffer in order to hold/take territory. Variation in value of these parameters leads to the following observed types of warfare:

Parameter	Observed Behaviour
$\alpha_{dT} \longrightarrow 1$	Trench War (Verdun)
$\alpha_{dT} \longrightarrow 0$	Guerrilla War
$\alpha_{aT} \longrightarrow 1$	The Somme
$\alpha_{aT} \longrightarrow 0$	Fixing Operations[2]

Table 1.1: Parameters used in the Adaptive Dynamic Model of Combat.

All Epstein claims is that this model gives a *less crude* caricature of combat dynamics. Our aim is not to over simplify to such an extent that our model is seen as a caricature, rather we strive to distil the essence of combat and avoid cluttering with terms aimed at instilling a high degree of realism.

While mathematical biology can provide much conceptual inspiration, it is from the development of numerical methods and computer simulation techniques discussed in the next section that we derive most benefit. Development of partial differential equations used for modelling swarming/schooling behaviour also provides inspiration for constructing a more sophisticated method with the aim of addressing the spatial dynamics criticism of the LE type approach to combat modelling.

1.2.3 Biological Aggregation Models

Approaches to modelling biological aggregations, incorporating such terms as swarming, schooling or clumping, have garnered much interest in recent years and there has been a plethora of approaches developed of which [39, 40, 42, 12, 6] are just a few. While there are both Lagrangian and Eulerian approaches, we concentrate here on the developments made in the area of continuous or Eulerian approximations.

There are numerous reason for animals to aggregate. Forces responsible for the formation of swarms include both responses to external stimuli such as light, nutrient gradients (eg chemotaxis) or the presence of predators; and internal forces such as social interactions (attraction, repulsion, alignment [32]). However one of the key concepts of many swarming observations is the maintenance of an individual separation distance determined by the environmental conditions at that time. Most swarms exhibit a relatively constant interior density and well defined edges. There is great variation in the approaches to modelling the mechanisms by which this occurs and thus the level of abstraction or detail included. Continuous approaches usually represent these through the inclusion of some form of nonlocal interaction [27, 35, 36], however there are others which achieve the same behaviour through setting acceleration proportional to the density [55]. These non-local interactions are generally not assumed to span great distances. Grunbaum [13] also states that an animal's social responses are inherently non-linear and are dictated by the conspecifics within the animal's restricted sensory field. Individual-based models represent this concept in a similar manner [36], for example, the perception function of Viscido [57] decays with distance. Alignment of individuals plays a pivotal role in some instances however for simplicity's sake we restrict ourselves to considering only the internal or social interactions of attraction and repulsion.

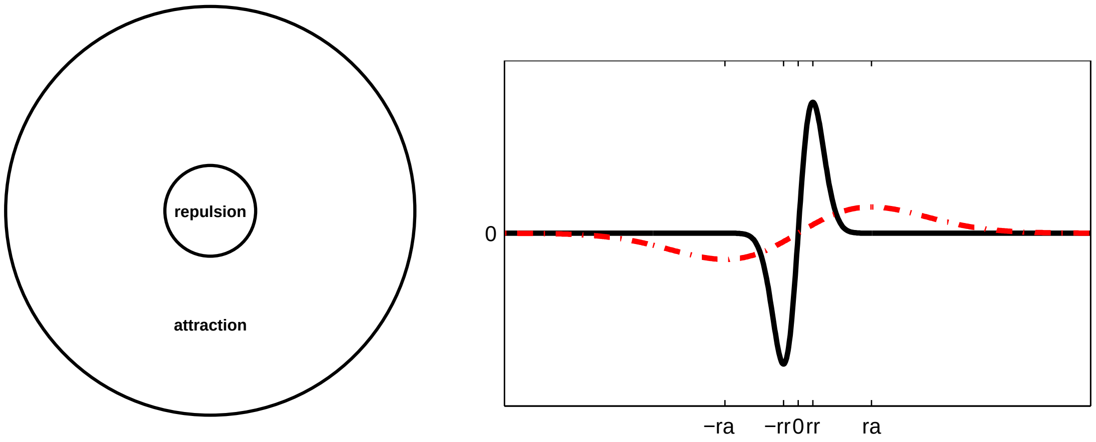
Interactions are taken to be pairwise [30, 56, 57] so that for a large population, these interactions are superimposed and are adequately described by a convolution for Eulerian models. Studies into non-local effects on swarming/schooling behaviour by Mogilner and Edelstein-Keshet [35], and Lee [27], used convolution terms to describe the effects of neighbours on movement. The form of the kernels used decayed with increasing separation distance, and the convolution domain was restricted to a finite range. This provides flexibility in defining the maximum distance at which a neighbour can be sensed thus providing a mechanism to model a range of sensors operational over differing ranges, be that visual, audible or otherwise.

Using the one dimensional convolution as an example:

$$K * u = \int_{x-r}^{x+r} K(x-X)u(X)\, dX \tag{1.10}$$

The kernel K associates the strength of the interaction with separation distance $x - X$. Mogilner and Edelstein-Keshet have found that a positive, even kernel ($K(x) = K(-x)$) on the interval $[-r, r]$ and zero otherwise gives rise to overall group drift, whereas an odd kernel ($K(x) = -K(-x)$) maintains the swarm profile without an overall drift. We also assume that the sensing and communication responsible for attractive and repulsive forces is omnidirectional. Thus an odd kernel is desirable as we do not wish the profile to move unless otherwise directed by a separate advection/velocity term, and it provides the correct phenomenological behaviour. Further expanding upon this idea, separating the effects of attraction and repulsion and by using an odd kernel, the convection term takes the form in one dimension:

$$\boldsymbol{C}(u) = \boldsymbol{V_u} + (A_a - A_r u)(K * u) \tag{1.11}$$



(a) Graphical representation of two dimensional attraction and repulsion ranges

(b) Form of the kernel in one dimension

Figure 1.2: Ranges and Kernels

The kernel K is odd as shown graphically in one dimension in Figure 1.2b, $A_a > 0$ and $A_r > 0$ represent the strength of attraction and repulsion and $\boldsymbol{V_u}$ $(a(e)_u$ in [35] and labelled the drift term) is the overall velocity term. In this form, attraction and repulsion are proportional to the *non-local* density. However repulsion is also proportional to *local* density such that for local densities greater than A_a/A_r repulsive effects dominate over attraction. Where density is low, repulsion has very little effect with attraction becoming the dominant term. Where density is high, attraction has little effect and repulsion prevents increasing overcrowding collapsing the aggregation to a point density. The two dimensional attraction and repulsion ranges are shown graphically in Figure 1.2a. The combined effect is a cohesive profile that has a constant interior density A_a/A_r with sharp edges [35].

We prefer to use the term velocity in preference to either advection or convection used in other fields where reaction/diffusion/advection systems are well known. This is due to the passive connotation of the term advection whereas soldier movement is perceived as deliberate. However all three terms are used interchangeably throughout this thesis.

1.3 Cellular Automata and Lagrangian Approaches

As mentioned in the Introduction, agent-based or cellular-automaton models have received much attention in defence related research into complex adaptive systems. As we wish to ultimately compare our continuous system of equations to results generated from a well known cellular automata model, it is worth noting that many presentations given regarding complex adaptive systems highlight the apparent chaotic and unpredictable behaviour demonstrated by the simplest cellular automata types - elementary cellular automata. We discuss what these are and the implications for our research.

1.3.1 Elementary Cellular Automata - ECA

So called as these one dimensional binary cellular automata are the simplest non-trivial case. The eight possible combinations produced by one cell plus its two neighbours, gives the 256 possible rule sets that define these ECA. These rule sets, usually numbered from 0 to 255, are capable of producing a range of visually seemingly complex and chaotic behaviours. Wolfram [58] has separated these ECA into four classes based on their observed behaviour:

Class	Observed Behaviour
I	Homogeneous state
II	Simple stable or periodic structures
III	Aperiodic or chaotic behaviour
IV	Long lived complex structures

Table 1.2: Classes of Cellular Automata as defined by Wolfram.

A well known example of a Class III type is Rule 30 and is often used as an example in cellular automata literature. This ECA can be described by 8 binary tuples:

$$(000, 001, 010, 011, 100, 101, 110, 111) \rightarrow (0, 0, 0, 1, 1, 1, 1, 0) \tag{1.12}$$

Despite the terms “chaotic” and “complex” used by Wolfram in his description of Class III ECA, research by Mingarelli *et al.* [9, 34] into continuous versions of cellular automata derived from a fuzzification process[3] has determined the evolution and dynamics of all elementary cellular automata. At most only nine of these fuzzy rules are truly chaotic with the remaining 247 rules deterministic. Their findings also highlighted the dangers of choice of visualisation method. For example in [9], depending on the chosen visualisation method for the evolution of the fuzzy ruleset derived from boolean rule 90, results may display behaviour remarkably similar to its CA counterpart, or may show the true rapid convergence to the asymptotic solution

[3] The disjunctive normal form which describes the given ruleset is fuzzified by redefining $(a \vee b)$ as $(a + b)$, $(a \wedge b)$ as ab and $(\neg a)$ as $(1 - a)$.

of $\frac{1}{2}$. This highlights the dangers of relying solely on particular visualisations and on one approach to derive conclusions.

The particular cellular automata model we will be comparing our model to will be described in more detail in Chapter 6.

1.3.2 Biological Modelling

There have been numerous CA or individual-based models derived for equally numerous biological systems - from the single-celled level of bacterial colonies to fish schooling to ungulate herds. Adopting this technique is a natural approach as the researcher is able to bestow individual members with individual characteristics and to more easily incorporate stochasticity. This also provides a valuable alternative to the more traditional deterministic Eulerian approaches. However, this is not to suggest either method is without fault and indeed it may be preferable to implement both methods so as to draw on the advantages of each.

For example, both Lagrangian and Eulerian approaches have been applied to the same biological system, elver movement, with a partial differential approach developed directly from an individual-based model (IBM) of that system [10]. Whilst the PDE was solved numerically, this approach was deemed to provide a method of analysing, extracting conclusions from (such as convergence to steady state solutions), verifying and communicating IBMs. With an increasing focus on IBMs, this type of procedure will become increasingly important.

Thus we believe there to be merit in pursuing the development and comparison of a continuous approach to a discrete one. While a detailed derivation from the cellular automata ISAAC is not undertaken here, the principle of using two different approaches to investigate the same dynamical system applies. Also, it is not the aim to develop a set of equations to replace Lanchester's equations and form an entirely new paradigm for the analysis of combat. Rather, the aim is to supplement the existing spatial extensions in order to ascertain a greater understanding of the underlying dynamics of these simplified combat models.

Chapter 2

New Approach

Here we develop a set of partial differential equations based on both Lanchester type equations and biological aggregation models. We retain the simple interaction term as developed by Lanchester and supplement this with the nonlocal convolution term as proposed by Protopopescu. This allows us to represent a range of weapons systems in a simple yet physically meaningful way as they take into account the local and non-local aspects of conflict.

We take the advances in biological aggregation modelling coupled with military requirements in order to develop a realistic method of modelling the spatial dynamics of soldier movement.

The terms of the equations will be referred to and treated as two separate groups - Spatial Dynamics and Interaction Terms. How each force navigates throughout the domain or battlespace is described by the Spatial Dynamics terms, with the Interaction Terms specifying the mechanisms through which forces interact with each other, largely through weapons firing.

2.1 Spatial Dynamics

One failing of Protopopescu's equations and published results is the gradual diffusion of the original Gaussian force density profile throughout the simulation. This did not pose a significant problem in Protopopescu's study [44], however, as the simulation time was so short that diffusion did not smear the profile to unrealistic sparse densities. However, the small, though significant, smearing present is not representative of actual troop movement. While some form of diffusion is intrinsic to animal aggregation [39], we expect some form of active diffusion showing density dependence to be present. This equates to some jostling within a force as it moves throughout the battle space, however the formation of a stable footprint or distribution with a sharp well defined edge is essential. As such, we use a density dependent diffusion term $\boldsymbol{D_u}$. For simplicity we maintain the assumptions that attraction and repulsion are omnidirectional, act over different ranges and disregard alignment of individual soldiers.

2.1.1 Military Considerations

Biological evidence suggests that in schooling and swarming behaviour, attraction acts over a greater spatial range than repulsion [27, 35, 48, 51], however both are limited to the limits of an animal's sensory perception. This assumption also holds in the military context. Each soldier within a formation has the benefit of an extensive communications capability such as personal radios to supplement basic human visual abilities, effectively extending the range of attraction out to the spatial size of the force itself. One purpose of these communications is for soldiers to maintain awareness of their own locations in relation to the rest of the force and to known enemy locations. Maintaining formation while traversing through thick jungle depending solely on the human senses is extremely difficult and thus the range of attractive forces in this context would be severely reduced. Supplementation with communications enhances the attractive range due to the basic human senses significantly. This is in keeping with the range of attraction extending over the range of the force itself.

Repulsion is deemed to operate over a much smaller range as this will depend on the proximity of a soldier's immediate neighbours. A soldier is required to maintain their position relative to the other soldiers within, usually, their view. Terrain and operational status will determine the density at any given time.

Thus we would expect that a single velocity term is inadequate for the military context, that is, separate attraction and repulsion kernels are required for a more realistic approximation of soldier movement.

We use the velocity term from Mogilner and Edelstein-Keshet with separate convolution term for attraction and repulsion:

$$\boldsymbol{C}(u) = \boldsymbol{V_u}u + A_a(K_a * u) - A_r u(K_r * u) \tag{2.1}$$

with:

$$K_a(x) = \frac{-x}{2a^2}\, exp\left(\frac{-x}{2a^2}\right), \quad K_r(x) = \frac{-x}{2r^2}\, exp\left(\frac{-x}{2r^2}\right), \tag{2.2}$$

where a represents the range of attraction, r the repulsion range and $a > r$

The approach developed here moves beyond well known advection-reaction-diffusion equations through the incorporation of non-local terms that represent cohesion of troops as they navigate the battle space.

2.2 Interaction Terms

Initially, the interaction between forces is represented using Protopopescu's formulation of two terms to represent aimed (short range weapons such as rifles) and area (long range weapons such as artillery) fire. For aimed fire, the terms take on a similar form to that seen in predator-prey modelling whereas the area fire term is a non-local convolution term.

2.2.1 Aimed Fire

Aimed Fire is represented by the terms for Force u and v respectively:

$$d_u v, \quad d_v u \tag{2.3}$$

Aimed fire is similar to the requirement of co-location of predator and prey in order for predation to occur. Here, troops must physically meet in order to affect each other. From the perspective of the numerical approximation, troops can interact if they occupy the same cell. Weapons such as rifles and grenades are represented in this way, as is hand-to-hand combat. While we recognise that these weapons, excluding hand-to-hand combat, could be represented by another more sophisticated non-local term, we believe that the simple terms proposed are sufficient to capture aimed fire behaviour and are in keeping with the overall level of abstraction of the Spatial Dynamics terms. All members of each force possess identical weapons, that is, are homogeneous with the respect to weapons capabilities.

In order to ensure positivity of results, the aimed fire terms may be extended to include a small positive term ϵ as was the case for Spradlin *et al.* [50]. Using Force u as an example, $d_u v$ becomes:

$$d_u v \frac{u}{\epsilon + u} \tag{2.4}$$

2.2.2 Area Fire

This term represents a distinct departure from the traditional Lanchester approach due to the spatial dependency. However typical commercially available wargames calculate area weapon effects in a similar manner. Usually the Linear Law (0.3,0.4) is seen as representing area fire, noting that there is no spatial dependency in the equation.[1] For simplicity, following Protopopescu's approach, we use a circularly symmetric function for the kernel over the domain R.

For Force u and v respectively:

$$u\left(k_u * v\right), \quad v\left(k_v * u\right) \tag{2.5}$$

We use the form $k(x,y) = \beta e^{-\nu\sqrt{|((x-X)^2+(y-Y)^2)-r_{op}|}} : \mathbb{R}^2 \rightarrow \mathbb{R}^+$ representing the strength of the interaction with separation distance $x - X$. Here, ν and β are positive weapons coefficients that represent the efficacy of the long range weapon with the optimal range r_{op} being modelled. This enables a wide range of long range weapons to be represented and will be referred to as the Artillery kernel. Other

[1] For example, JANUS is an interactive simulation wargame used extensively throughout the world's military [4]. During use, the simulation is periodically halted to allow military participants to discuss developments and alter the missions of their forces. Once the simulation is restarted, kill probabilities based on type and separation distance are calculated and enforced. JANUS also uses three dimensional terrain data in loss calculations although in simpler wargames, these probabilities are in the form of a simple equation relating weapon type and distance.

forms of the kernel such as a bivariate Gaussian may be used [50]; however we believe this form adequately captures area weapon behaviour. We also use another form of the kernel $k(x,y) = \beta e^{-\nu\sqrt{|(x-X)^2+(y-Y)^2|}}$ for use in the cellular automata comparison scenarios, to represent weapons such as heavy guns that have a range greater than the individual soldier's reach, and are also decreasingly effective with decreasing distance from the firer. Depending upon the choice of weapon system to be represented, either of these two kernels may be used. Replacing the aimed fire term with this second Rifle kernel would also be useful in many scenarios. Due to the dependency of this term on both force densities, positivity is ensured.

2.2.3 Other Terms

While reinforcement rates can be critical in some circumstances, we elect to exclude these terms for two reasons: (i) simplicity and (ii) the model we are striving to make comparisons with does not include reinforcements.

2.3 New Equations

Combining the new spatial dynamics and interaction terms, we propose the following form of the integro-differential equations in two dimensions. [We separate each equation into three parts, $\mathbf{f_{diff}}$ containing the diffusion terms, $\mathbf{f_{vel}}$ containing the velocity terms and $\mathbf{f_{react}}$ containing the interaction terms for ease of discussion in the following sections,]

$$\frac{\partial u}{\partial t} = \underbrace{\nabla \cdot (\boldsymbol{D_u}(u)\nabla u)}_{\mathbf{f_{diff}}} + \underbrace{\nabla \cdot \{u(\boldsymbol{V_u}u + A_a(K_a * u) - A_r u(K_r * u))\}}_{\mathbf{f_{vel}}} + \underbrace{u\,(k_u * v) + d_u v}_{\mathbf{f_{react}}} \tag{2.6}$$

$$\frac{\partial v}{\partial t} = \underbrace{\nabla \cdot (\boldsymbol{D_v}(v)\nabla v)}_{\mathbf{f_{diff}}} + \underbrace{\nabla \cdot \{v(\boldsymbol{V_v}v + A_a(K_a * v) - A_r v(K_r * v))\}}_{\mathbf{f_{vel}}} + \underbrace{v\,(k_v * u) + d_v u}_{\mathbf{f_{react}}} \tag{2.7}$$

We consider a rectangular domain Ω with lengths L_x and L_y. The general form of the boundary conditions for the domain $(0, L_x) \times (0, L_y)$ is:

$$a_{11}u + b_{11}\frac{\partial u}{\partial x} = c_{11} \quad \text{at } x = 0$$

$$a_{21}u + b_{21}\frac{\partial u}{\partial y} = c_{21} \quad \text{at } y = 0$$

$$a_{12}u + b_{12}\frac{\partial u}{\partial x} = c_{12} \quad \text{at } x = L_x$$

$$a_{22}u + b_{22}\frac{\partial u}{\partial y} = c_{22} \quad \text{at } y = L_y$$

Boundary conditions may be either:

$$\text{Dirichlet} \quad (a_{ij} \neq 0, b_{ij} = 0, \forall i = 1, 2; j = 1, 2),$$

$$\text{Neumann} \quad (a_{ij} = 0, b_{ij} \neq 0, \forall i = 1, 2; j = 1, 2) \quad \text{or}$$

$$\text{Mixed} \quad (a_{ij} \neq 0, b_{ij} \neq 0, \forall i = 1, 2; j = 1, 2).$$

Variable	Definition of Variable
$u(x, y, t)$	The positive troop density of Force u at the given position and time.
$v(x, y, t)$	The positive troop density of Force v at the given position and time.
Parameter	Definition of Parameter
$D_i(x, y, t)$	The diffusion coefficient for Force $i(i = u, v)$ that may be spatially, temporally or otherwise dependent.
$V_i(x, y, t)$	The velocity coefficient for Force $i(i = u, v)$ that may be spatially, temporally or otherwise dependent.
A_{a_i}	The strength of the internal attraction of Force $i(i = u, v)$.
A_{r_i}	The strength of the internal repulsion of Force $i(i = u, v)$. The ratio of $\frac{A_{a_i}}{A_{r_i}}$ gives the desired internal density of Force i.
K_{a_i}	The form of the attraction kernel for Force $i(i = u, v)$. For the results presented here, the form is $K_a * u = \int_{y-r_a}^{y+r_a} \int_{x-r_a}^{x+r_a} K_a(x - X, y - Y)u(X, Y)dXdY$
K_{r_i}	The form of the repulsion kernel for Force $i(i = u, v)$. For the results presented here, the form is $K_r * u = \int_{y-r_r}^{y+r_r} \int_{x-r_r}^{x+r_r} K_r(x - X, y - Y)u(X, Y)dXdY$
r_{a_i}	The range over which the attraction kernel for Force $i(i = u, v)$ acts.
r_{r_i}	The range over which the repulsion kernel for Force $i(i = u, v)$ acts. $r_{r_i} < r_{a_i}$
k_i	The form of the area fire kernel for Force $i(i = u, v)$. For the results presented here, the form is $k_i(x, y) = \beta_i e^{-\nu_i \left\| \sqrt{(x-X)^2+(y-Y)^2} \right\|} : \mathbb{R}^2 \to \mathbb{R}^+$ for the rifle kernel, and $k(x, y) = \beta e^{-\nu \sqrt{\|((x-X)^2+(y-Y)^2)-r_{op}\|}} : \mathbb{R}^2 \to \mathbb{R}^+$ for the artillery kernel.
r_{af_i}	The range over which the area fire kernel for Force $i(i = u, v)$ acts.
r_{op_i}	The optimal range for the area artillery fire kernel for Force $i(i = u, v)$.
β_i	Variable used in the calculation of area fire losses for Force $i(i = u, v)$.
ν_i	Variable used in the calculation of area fire losses for Force $i(i = u, v)$.
d_i	The aimed fire coefficient for Force $i(i = u, v)$. To ensure positivity, the form of the term should be as given in Eq. eq:PosAimedFire.

Table 2.1: Definitions of Variables and Parameters Used in Our PDE Model.

Attraction and repulsion operate over different spatial domains, r_a and r_r respectively with $r_a > r_r$, and the drift term is replaced by a spatially dependent vector field V. So now the overall movement throughout the domain and cohesion of the soldier formation is represented in (2.6)-(2.7). The traditional convection/diffusion/reaction equations have now been altered to provide a more realistic representation or force movement throughout a battlefield.

This system of equations will be further expanded in order to more closely mimic the functions used within ISAAC as outlined in Chapter 6.

Chapter 3

Numerical Techniques

In this chapter we discuss the general method of classification of partial differential equations, typical techniques used for numerically approximating them, and their advantages/disadvantages.

Numerical methods used in previous combat models are then discussed, highlighting the difficulties of using these techniques with our new approach by demonstrating the types of numerical errors found.

Finally we outline a numerical method that has been applied to biological systems such as in the approximation of fungal growth, which fulfils our requirements of conservation of density while maintaining good resolution, mainly due to the similarity of the equations in question. We then summarise the numerical method to be used in this research.

3.1 General Form of Partial Differential Equations

One of the most general ways of writing a second order linear partial differential equation is the form:

$$A\frac{\partial^2 u}{\partial x^2} + B\frac{\partial^2 u}{\partial x \partial y} + C\frac{\partial^2 u}{\partial y^2} + \ldots = 0 \tag{3.1}$$

By using this notation, three common types of PDEs can be defined:

Hyperbolic $B^2 - 4AC > 0$

Parabolic $B^2 - 4AC = 0$

Elliptic $B^2 - 4AC < 0$

Whilst some simple PDEs may be solved exactly, there may be no easy way to analyse other types due to the presence of nonlinearities in the remaining terms in (3.1). As analytical methods are unavailable, numerical approximations can be made which usually entails the use of computer simulations. In many fields such as mathematical biology, the interest lies in time evolution and asymptotic states. Usually these are some type of initial value problem with conditions at some time t_0 specified, along with domain and boundary conditions.

The most common three techniques used for numerical approximation are the Finite Difference, Finite Volume, and Finite Element Methods. The basis of all three of these methods is the discretisation of the domain in order to approximate the spatial derivatives and to then evolve the equation for a discrete time interval. Typically both the spatial and temporal domains are uniformly discretised.

How these discretisations are used to evolve the equations can be broadly classed as either explicit or implicit. An explicit method requires only the current values of the function in order to calculate the values at the next time step whereas an implicit method uses both current and future values.

For a finite difference scheme, derivatives are approximated at the mesh points and are usually separated into three types - forward, backward or central. Finite Volume methods involve evaluating the fluxes at the boundaries of each cell surrounding each mesh point. This type of method has the added advantages of the flexibility of using non-uniform meshes, and conserving density as the fluxes across boundaries cancel.

Depending upon the form of the equation(s), one or more numerical technique(s) may be suitable. Here we give a few examples of typical forms of these equations and common numerical methods that may be used.

3.1.1 Parabolic

Another typical form of a parabolic equation is:

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(b(x,t)\frac{\partial u}{\partial x}\right) + c(x,t)u + d(x,t). \quad (b > 0)$$

One of the simplest approximations that can be used is a combination of explicit finite difference schemes for the time and space discretisations. For example, a forward time and second order spatial central difference method or “FTCS” scheme. The Crank-Nicolson method uses a central difference time and second order central difference spatial derivative and is a well known example that uses implicit methods. Under certain conditions these methods can be numerically stable and convergent, however they may be computationally expensive and introduce unwanted errors for nonlinear parabolic equations. Explicit methods are generally the least computationally expensive yet require small time steps to maintain accuracy. Implicit methods are more computationally expensive, however, allow for larger time steps to be taken.

3.1.2 Hyperbolic

A typical form of a scalar hyperbolic equation is

$$a\frac{\partial u}{\partial t} + b\frac{\partial u}{\partial x} = c \quad (a, b \neq 0).$$

Using the example of linear advection,

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0, \tag{3.2}$$

solutions can be found along *characteristics*. These curves are solutions of the ordinary differential equation

$$\frac{dx}{dt} = a(x, t). \tag{3.3}$$

In general the method of characteristics uses these equations and initial data $u(x, 0)$ to find a series of curves which satisfy $u(x, t) = u(x_j, 0)$ where $x_0, x_1, \ldots, x_j, \ldots$ is a suitably chosen set of points. Difficulties arise for nonlinear hyperbolic equations where these characteristics overlap, that is, there exist multiple values for the solution which can be viewed as the formation of a discontinuous shock wave. The classic equation used for demonstrating this phenomenon is the inviscid Burgers Equation,

$$\frac{\partial u}{\partial t} + \frac{\partial \frac{1}{2} u^2}{\partial x} = 0. \tag{3.4}$$

Generally none of the previously mentioned schemes are capable of providing sufficient accuracy around the shock as well as other areas. Thus we must look for other methods or a combination of several methods in order to provide the desired accuracy.

3.1.3 Elliptic

Elliptic equations are generally time independent equations describing physical phenomena such as Poisson's equation, $\nabla^2 u = f(x)$. As these equations are not of the relevant type, we recommend references such as [29] for further reading.

3.1.4 Mixed-type

Reaction-advection-diffusion equations are of the mixed type - both hyperbolic and parabolic - with advection dominated reaction-diffusion equations abundant throughout science and nature. One of the most famous examples of which is the chemical reaction-diffusion system that mimics the development of a leopard's spots developed initially by Alan Turing [53] and then expanded by Sy-Sang Liaw and Ruey-Tarng Liu [31]. Other examples such as the Belousov-Zhabotinski reaction [2], models of fluid flows or turbulence show the intricate detail that must be captured in order to adequately demonstrate and understand their underlying dynamics. These equations often possess steep fronts, shocks or discontinuities whose resolution must be addressed accurately in order to understand the characteristics of the system under investigation. This remains one of the foremost challenges for the development and use of numerical approximations.

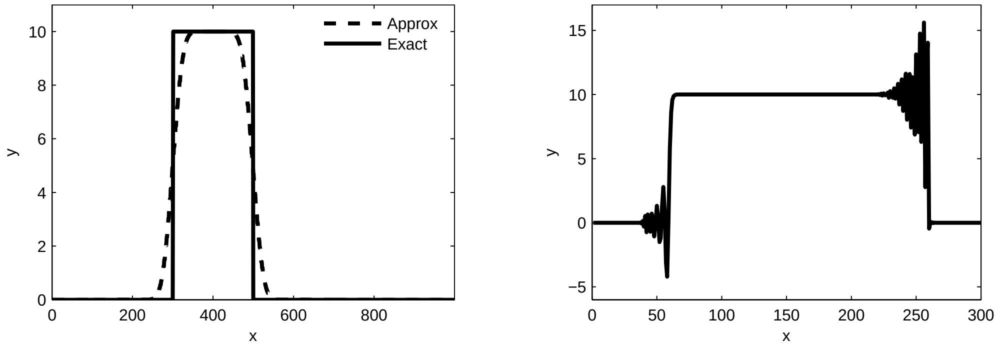
Due to the mixed nature of reaction-advection-diffusion equations, we look to more sophisticated techniques. Examples of commonly used methods for solving include Multiscale, Galerkin, Upwind Petrov-Galerkin, Method of Lines, Finite Volume and Flux Limiter methods (see [29] for an excellent review of numerical methods). It is thanks to the abundance of these equations throughout nature that such diversity of numerical methods can now be found throughout the literature.

As our equations are of the mixed-type, we will look to implement a combination of these methods.

3.2 Previously Used Numerical Methods

We begin by describing the numerical methods employed in the previous combat modelling by Protopopescu and Jaiswal and noting the deficiencies of these and other well known yet simple techniques.

Application of standard simple numerical methods can result in a range of undesirable behaviours due to their inherent inadequacies for use unmodified with non-linear PDEs. When the convection field dominates over diffusion or if the grid spacing is too large, excessive unrealistic oscillations or *numerical oscillation* as seen in Figure 3.1b can occur. An example of this is seen when using a second order centred finite difference method for a scalar advection equation. As moving fronts may exist, the integrator approximation method, grid spacing and time step must be chosen with care otherwise further errors may be introduced. Again using the example of scalar advection, a first order two point upwind finite difference method can result in *numerical diffusion* as seen in Figure 3.1a.



(a) Numerical Diffusion seen when approximating scalar advection.

(b) Numerical Oscillation around discontinuities.

Figure 3.1: Examples of possible errors in some Numerical Techniques.

Protopopescu recognised the difficulties in numerical approximations of the Central Finite Difference scheme used when approximating sharp fronts [47], and so used sufficiently smooth force profiles in order to avoid these difficulties. Spatially dependent velocity fields were not investigated due to unrealistic losses through boundaries, and density dependency was also omitted. The approximation method used by Protopopescu was the Method of Lines (MOL). This technique involves spatially discretising the PDEs so producing a set of ordinary differential equations continuous in time. A suitable integration technique is then applied. He began by specifying the allowed error of time integration of the ODE system, then investigated mesh spacings that resulted in a spatial error of equal or lesser magnitude. Protopopescu and Jaiswal did not use more advanced techniques such as the adaptive MOL, upwinding, or mesh refinement, as a regular grid spacing was deemed to suffice.

However the application of an incorrect more advanced technique does not guarantee success with deficiencies demonstrated by the following example. Manipulating code based on the upwinding finite difference scheme of Schiesser [49] as developed by Lee *et al.* [28], a Matlab model of (2.6) and (2.7) was developed. Each force is moved with constant velocity in the direction of the opposing force's initial position and velocity is not altered during the simulation to account for pursuance as demonstrated by Protopopescu. Diffusion was also set as constant. Initial force profiles were of the form:

$$u(x,0) = \frac{size}{0.05 * \sqrt{2\pi}} e^{\frac{-((x-\mu)^2)}{2\rho^2}}$$

Details of the numerical method as used by Lee are as follows. A method of lines is used with a selection of fourth order spatial approximations to the derivative of the function f given by the matrix:

$$B = \frac{1}{24\Delta x}\begin{bmatrix} -50 & 96 & -72 & 32 & -6 \\ -6 & -20 & 36 & -12 & 2 \\ 2 & -16 & 0 & 16 & -2 \\ -2 & 12 & -36 & 20 & 6 \\ 6 & -32 & 72 & -96 & 50 \end{bmatrix} \tag{3.5}$$

In Equation 3.5, the middle row represents the central fourth order spatial approximation, the first and second rows represent fourth order backward difference approximations and the fourth and fifth rows represent fourth order forward approximations. This matrix is used to calculate derivatives on the lattice where for example, the central difference approximation to u' is given by:

$$\frac{\partial u}{\partial x} = \frac{1}{24\Delta x}\begin{bmatrix} 2 & -16 & 0 & 16 & -2 \end{bmatrix} \odot \begin{bmatrix} f_{i-2} \\ f_{i-1} \\ f_i \\ f_{i+1} \\ f_{i+2} \end{bmatrix} \tag{3.6}$$

Neumann boundary conditions of zero flux were used. Other parameters used were $\Delta x = 0.01, \Delta t = 0.2, D_u = 0.005, D_v = 0.008, C_u = 1, C_v = -1.5, size_{u,v} = 2000, \rho_{u,v} = 0.05, \mu_u = 3, \mu_v = 7, d_u = 3 \times 10^{-4}, d_v = 2.4 \times 10^{-4}, \beta_u = 3 \times 10^{-4}, \beta_v = 2.4 \times 10^{-4}, \nu_{u,v} = 0.3$.

After each time step which was calculated using the inbuilt Matlab integrator **ode45**, interaction terms were solved to reduce force density which then produced the input density profile for the following time step calculation. In effect troop movement for the given time step is conducted, and based upon these updated distributions casualties are determined and deducted from the distributions. For initial modelling, strategic decisions such as retreat or advance were omitted in order to investigate the behaviour of the model and ensure accuracy of the results. The results are seen in Figure 3.2 where the smearing effect of diffusion is clearly demonstrated. Both forces advance regardless of the relative position of the opposing force or density loss. The top graph shows the superimposed distributions of both forces at the specified times, with the middle two graphs showing the individual force distributions for extra clarity.

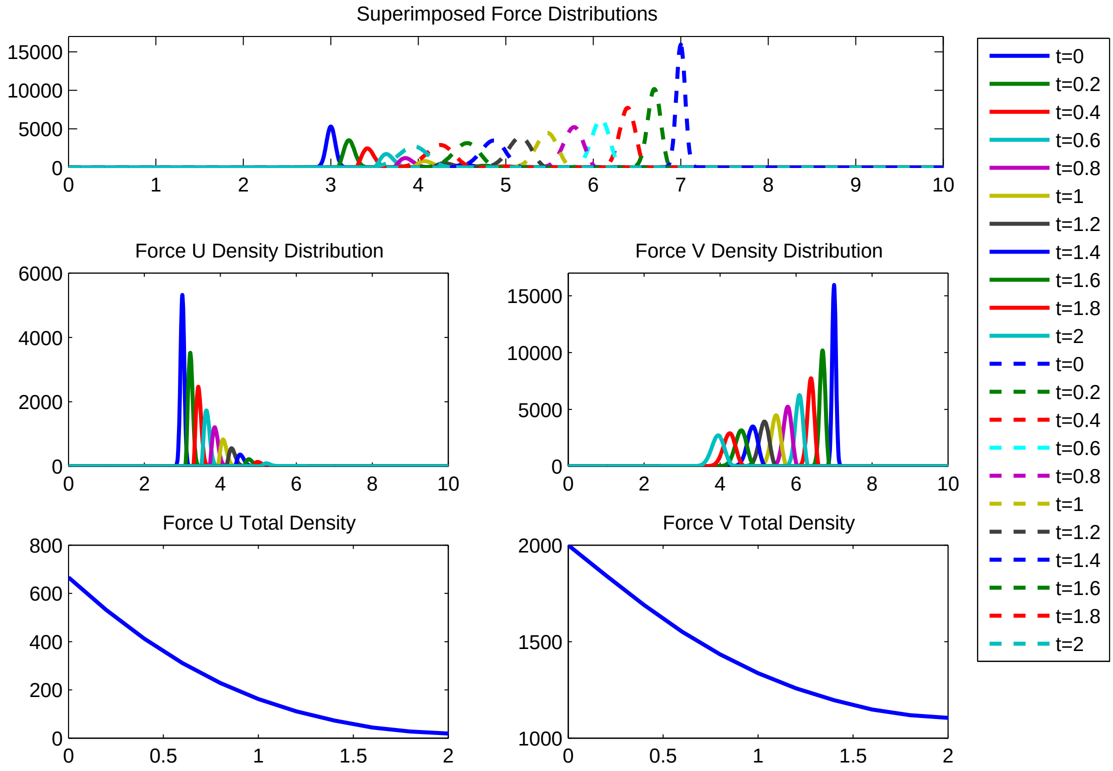


Figure 3.2: Basic example using MOL with Aimed Fire Only. Despite a small value of diffusion, the initial force profiles rapidly and unacceptably diffuse.

Higher strategic decision making abilities were then included. In Figure 3.3 a retreat mechanism for both forces is specified to be enforced once a threshold density loss is encountered. This threshold is lower for Force u and is encountered sooner than that of Force v such that v does not retreat throughout the duration of the simulation. Again the top graph shows the superimposed distributions of both forces at the specified times, with the middle two graphs showing the individual force distributions for extra clarity.

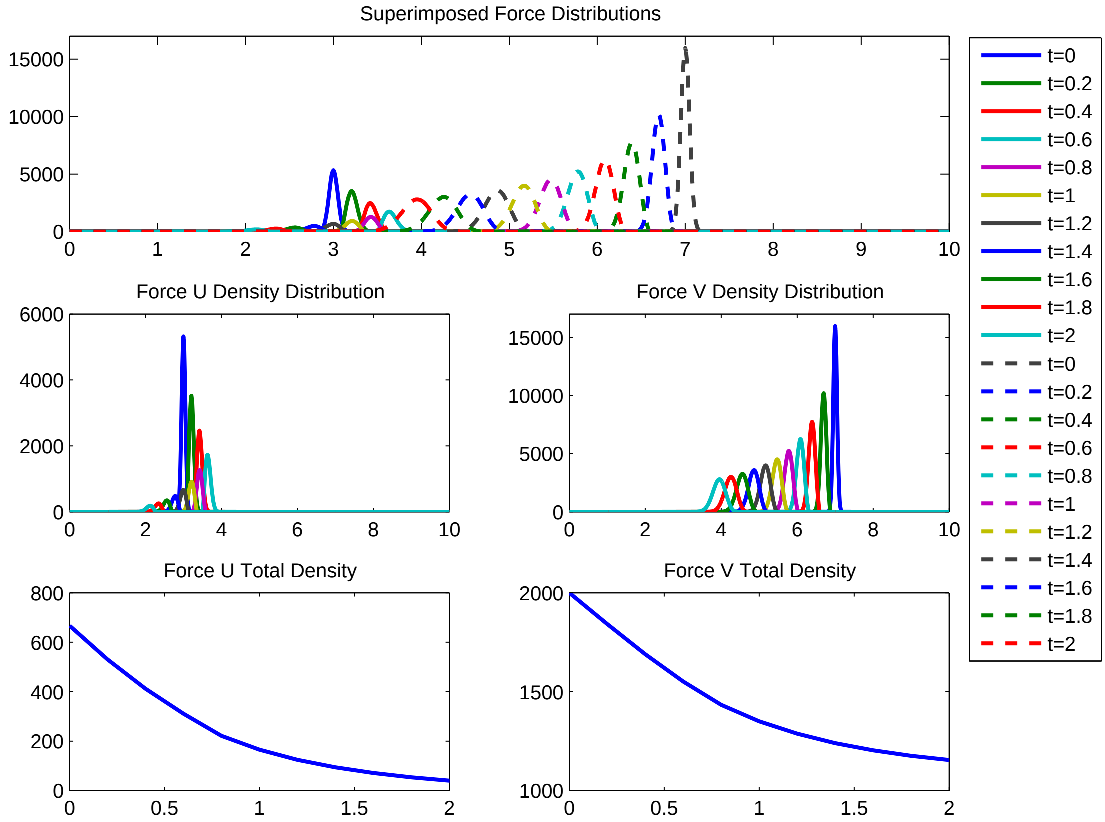


Figure 3.3: Basic example using MOL with Aimed Fire Only, and the addition of retreat decision criteria for both Forces. Force u reaches its threshold of 20% losses before triggering a retreat whereas Force v does not reach its 50% threshold

Extending this one dimensional model into two dimensions using the same method of lines upwinding finite difference scheme by Schiesser [49] as detailed in Equations 3.5 and 3.6, created numerical abnormalities or striations when a high value of diffusion was used. Using the **ode15s** integrator as suggested in the Matlab literature instead of **ode45** as used in the one dimensional case did not resolve this issue. Advection only did not produce these striations.

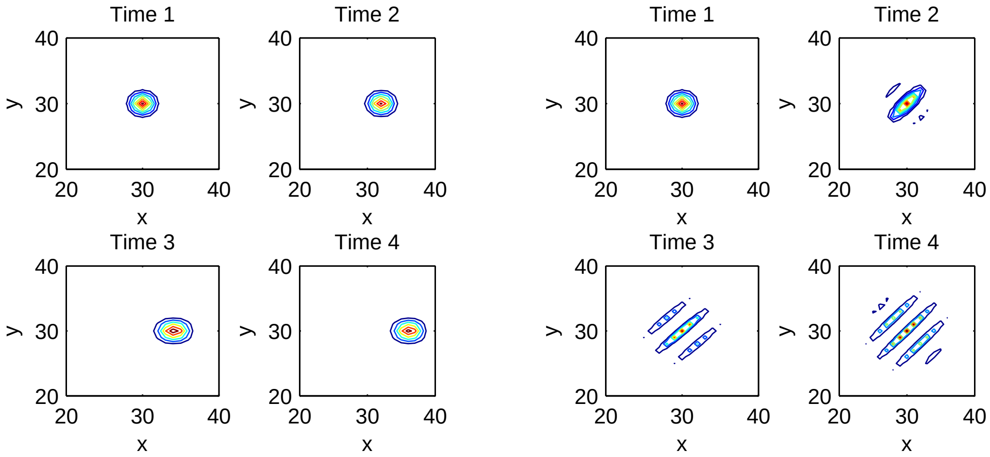


Figure 3.4: Integrator errors present with high values of diffusion (left) and absent with only advection (right).

Changing the form of the advection term from a simple constant to, for example, a density dependent term highlighted a further difficulty of using inbuilt Matlab integrators. Conservation of troop density in the absence of attrition was not maintained as the integrators trialled are not symplectic. Thus this technique coupled with Matlab integrators was not found to be a suitable approach to this problem.

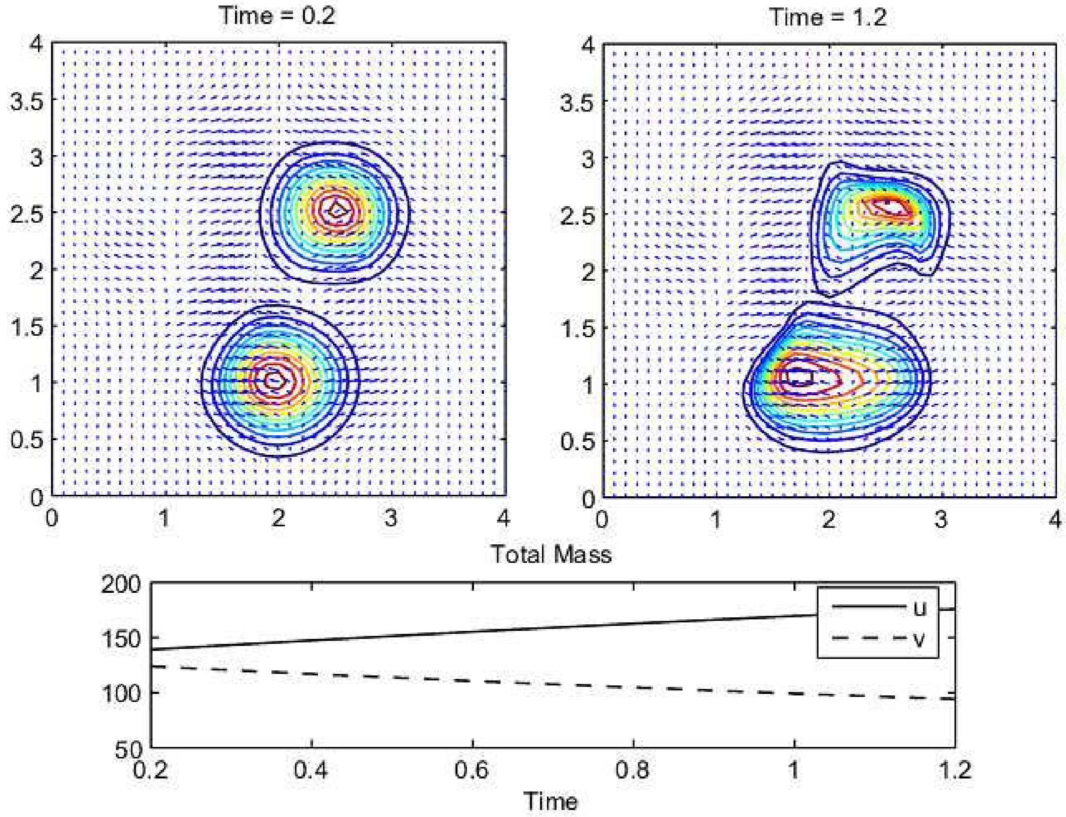


Figure 3.5: Total Mass, Spatially Dependent Advection

While well known methods such as Lax-Wendroff have been successfully implemented for simpler versions of (2.6) and (2.7) (see [35]), an extension of this method to incorporate density dependent diffusion leads to spurious unphysical oscillations, a well known failing of the Lax-Wendroff technique. Nonphysical negative values are unacceptable and must not be a feature of the numerical method used to approximate these equations. If higher order accurate methods are to be used, some form of flux limitation is needed to ensure physically acceptable results.

As a non-constant field will be required for both velocity and diffusion in further modelling, we must now look to other numerical methods such as conservation laws that have been widely utilised in other areas such as mathematical biology and gas dynamics.

3.3 Conservation Laws

The standard conservation form of a vector of unknown functions $u(x,t)$ and a vector of flux functions $f(u)$ is:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\partial x} = 0. \tag{3.7}$$

As this system of coupled partial differential equations describes the evolution of the density of troops, the extra demands of mass conservation and of positivity further restricts the choice of numerical method. Conservation laws enable accurate capturing of discontinuities and shock speeds, and ensure, by definition, conservation of density. For mixed-type equations such as these, with a hyperbolic (advection) and parabolic (diffusion) component, there exists only a handful of appropriate techniques suitable for use in solving them: symmetrisation, upwinding, exponential fitting and least-squares regularisation [37]. As no one technique is suitable for all problems, the researcher must decide those features of the equation(s) in question are critical which will then determine the choice of technique.

We follow the work of Gerisch [1], Boswell [3] and Hundsdorfer [16] by using the Method of Lines scheme. Here spatial derivatives are discretised on a uniform finite grid $x_i, i = 0, 1, \cdots, N$ with spacing Δx, giving a large system of Ordinary Differential Equations. Flux limiters are employed at this stage to ensure positivity and conservation of mass. An explicit Runge-Kutta method is then used for the time integration, again with constraints in place to ensure positivity.

3.3.1 Spatial Discretisation

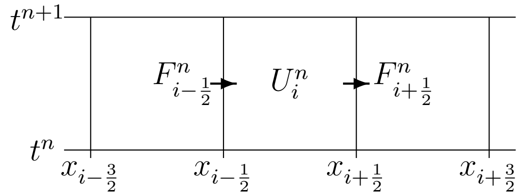


Figure 3.6: Flux Calculation for One Dimension

Following standard finite volume theory, the central difference approximation in one dimension to the spatially discretised conservative form of the equation $\frac{du}{dt} = F(x,t),\ t \geq 0,\ u(x,0) = u_0(x) \geq 0$ takes the form:

$$\frac{d}{dt}U_i + \frac{\left\{F_{i+1/2} - F_{i-1/2}\right\}}{\Delta x} = 0,\ i = 0, 1, \cdots, N. \tag{3.8}$$

where U_i is the continuous time approximation of $u(x_i, t)$ over cell i at time t. At time n, the density function is defined by

$$U_i^n = \left(\frac{1}{\Delta x}\right) \int_{x-1/2}^{x+1/2} u(x, t^n) dx \quad i = 0, 1, \cdots, N. \tag{3.9}$$

The general flux function F is defined as a function of the fluxes of the surrounding grid points. Let the flux at each grid point be defined as:

$$f_i(t) = v(x_i, t)U_i(t) \tag{3.10}$$

Let the general flux function for positive velocities become:

$$F_{i+1/2} = f_i + \frac{1}{2}\Phi(r_{i+1/2})(f_i - f_{i-1}) \tag{3.11}$$

Similarily for negative velocities:

$$F_{i+1/2} = f_{i+1} + \frac{1}{2}\Phi(r_{i+3/2}^{-1})(f_{i+1} - f_{i+2}) \tag{3.12}$$

where the limiter function Φ is a function of the smoothness of the fluxes as measured by r:

$$r_{i+1/2} = \frac{f_{i+1} - f_i}{f_i - f_{i-1}} \tag{3.13}$$

This limiter function may act as a switch between high and low order approximations. Smooth areas can then be approximated using a higher order scheme whereas oscillations that might otherwise be present in areas of steep gradients can be approximated using a lower order scheme. Based on [1, 3, 16] which specify requirements for maintaining positivity, we use the van Leer limiter:

$$\Phi(r) = \frac{\|r\| + r}{1 + \|r\|} \tag{3.14}$$

In the numerical simulations and following, a small number $\epsilon \ll 1 = 10^{-30}$ is added to both numerator and denominator of (3.13) to ensure good behaviour in regions of uniform flow [1].

Now that we have a system of ODEs with initial conditions, the appropriate time integration method must be determined. An explicit three stage second order Runge-Kutta scheme with matrix A and weights b is used that provides good positivity and second order accuracy [1].

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 1/2 & 0 & 0 \\ 1/2 & 1/2 & 0 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix} \tag{3.15}$$

Again following Boswell, a variable time step is used in order to maintain accuracy without unnecessarily small time steps. Using the initial values u_k at time $t = t_k$, two approximations are found at $u(t = t_k + \tau)$. Firstly, using the single time step τ and secondly using two consecutive time steps of $\tau/2$. A scaled error is defined by the standard measure:

$$\rho = \frac{1}{2^p - 1}\sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\frac{(u_{(2\times\frac{\tau}{2})})_i - (u_\tau)_i}{atol_i + rtol_i|(u_k)_i|}\right)^2} \tag{3.16}$$

where $p = 2$ is the order of the ODE solver, $m = N$ is the ODE system dimension, and $atol_i$ and $rtol_i$ the respective absolute and relative tolerances of the error. If $\rho > 1$ the step is rejected, and it follows that if $\rho \leq 1$ the step is accepted. Whether the step is recomputed or not, the new time step is calculated as:

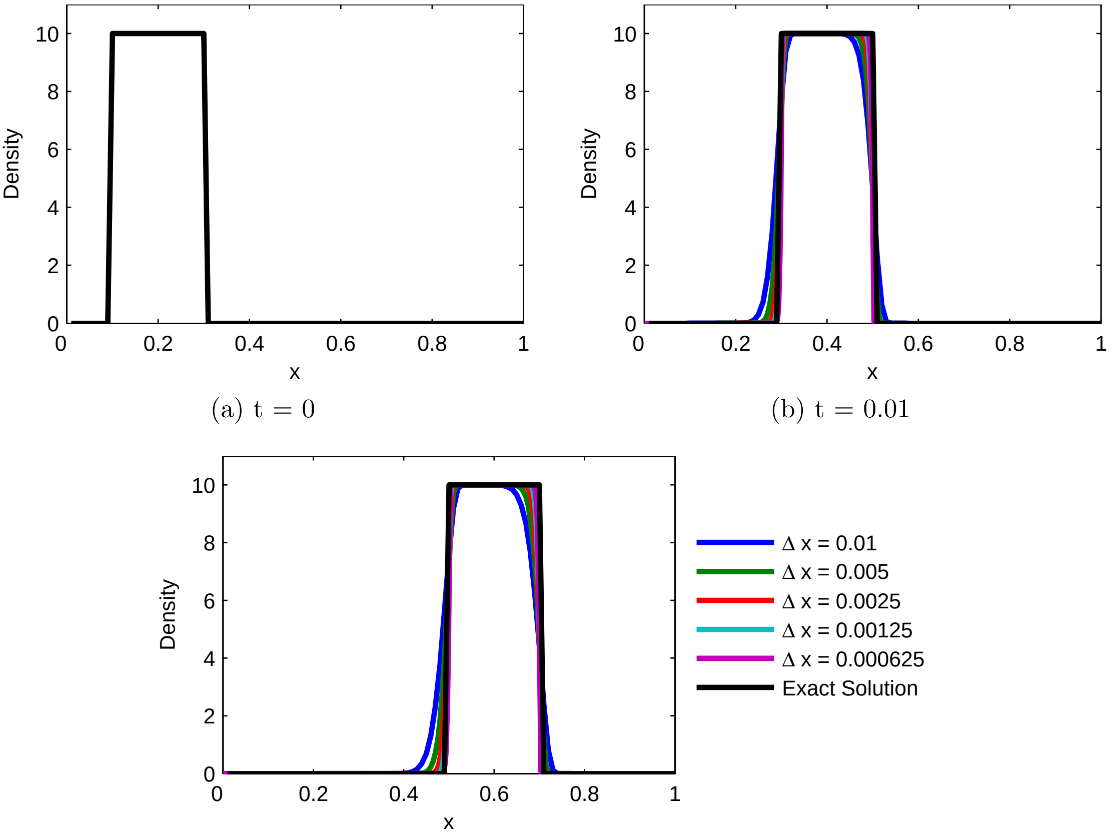
$$\tau_{new} = \tau \cdot min\{2, max\{0.8\rho^{\frac{-1}{2^p-1}}, 0.25\}\} \tag{3.17}$$

Once the step is accepted, the new state is updated using the more accurate approximation $u_{2\times\frac{\tau}{2}}$.

3.3.1.1 One Dimensional Test Case

In order to demonstrate the ability of this numerical method to resolve discontinuities and following standard demonstrations in numerical methods texts [29, 33], the numerical approximation of several limiters is compared to the exact solution for the constant advection of a rectangular function for various times as shown in Figure 3.7. Firstly, the one dimensional case using the van Leer limiter with velocity $C = 20, \Delta t = 10^{-6}, atol = rtol = 10^{-6}$ and the initial density profile

$$u(x,0) = \begin{cases} 10 & 0.1 \leq x \leq 0.3 \\ 0 & \text{otherwise} \end{cases}$$



(a) t = 0

(b) t = 0.01

(c) t = 0.02

Figure 3.7: Numerical approximation comparison to exact solution of advecting rectangular function.

As Figure 3.7 shows, with decreasing grid spacing Δx the resolution of the leading and trailing edges of the rectangular function increases. The correct location of these edges is also maintained and most importantly the solution remains positive throughout. There are no negative oscillations present.

3.3.1.2 Two Dimensional Test Case

Now expanding to the two dimensional case again using the van Leer limiter with velocity $C_{x,y} = 20$ and the initial density profile

$$u(x,y,0) = \begin{cases} 10 & 0.1 \leq x \leq 0.3 \text{ and } 0.1 \leq y \leq 0.3 \\ 0 & \text{otherwise} \end{cases}$$

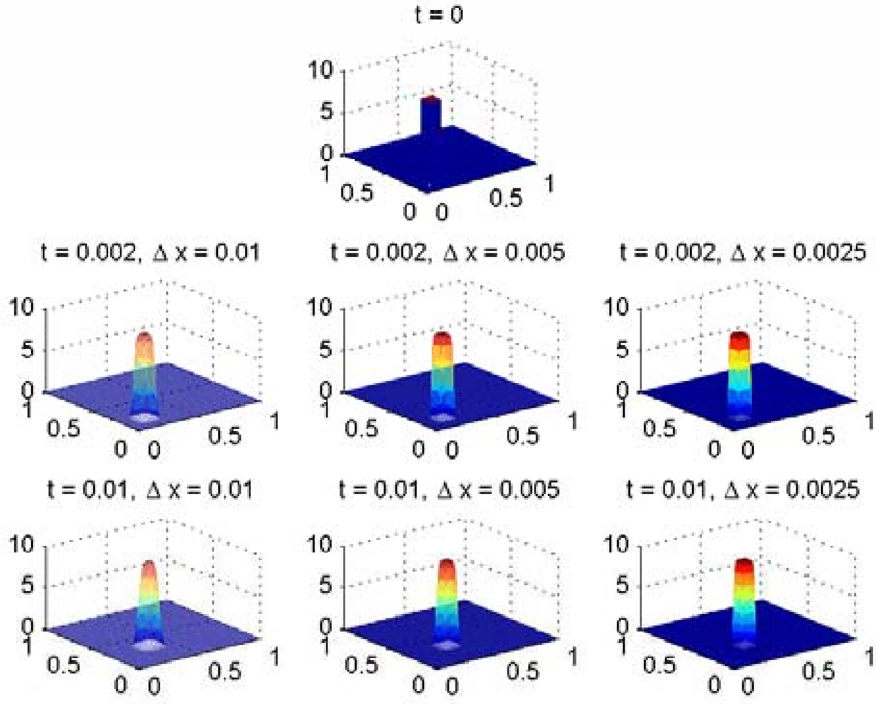
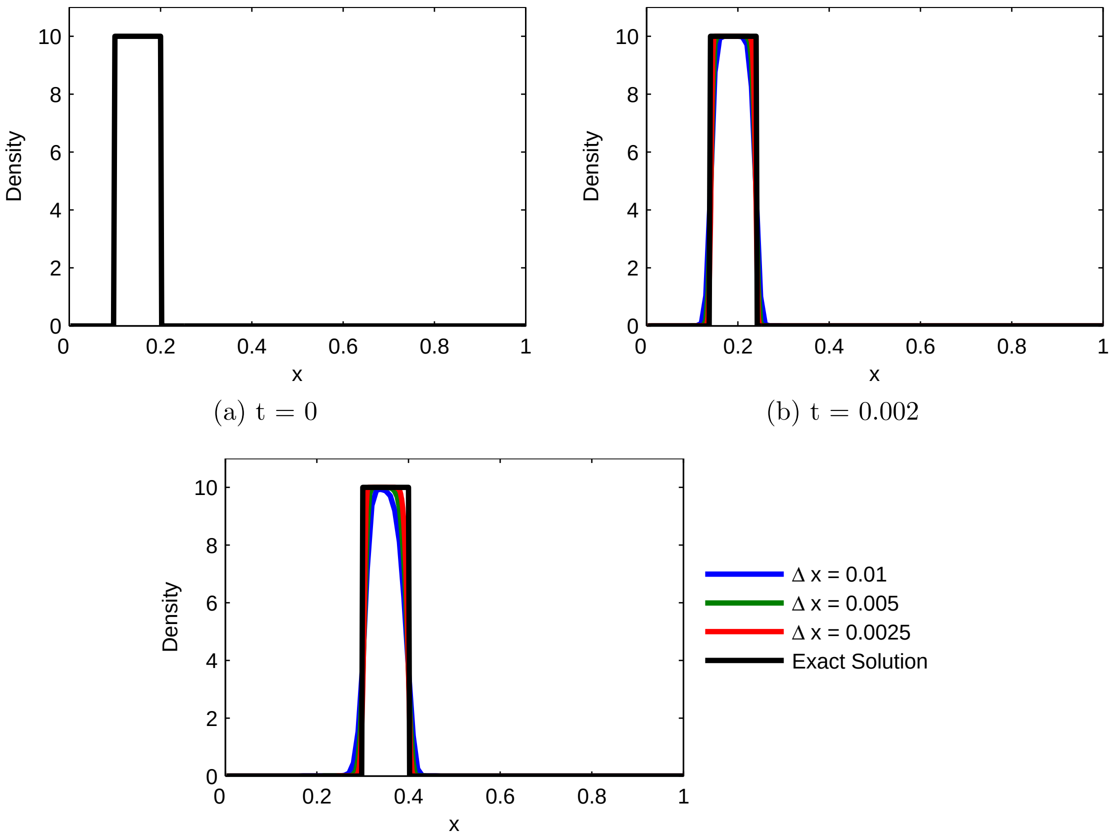


Figure 3.8: Numerical approximation comparison to exact solution of advecting square function.



(a) t = 0

(b) t = 0.002

(c) t = 0.01

Figure 3.9: Numerical approximation comparison to exact solution of advecting square function, diagonal slice.

Again Figure 3.8 and the two dimensional diagonal slice of Figure 3.9 shows that with decreasing grid spacing the resolution of the edges of this square function increases and the correct location of these edges is maintained. Again it can be seen that at no stage are there negative parts to the approximations arising from oscillations - the solution remains positive throughout. Although a number of different limiters can be used, the van Leer limiter is implemented here and as will be shown in the following chapter, this is the most appropriate choice of limiter.

3.3.2 Numerical Integration

We apply the numerical method above for the numerical integration of the advection section in (2.6) and (2.7). For the density dependent diffusion component of (2.6) and (2.7), we use a simple second order accurate finite difference scheme:

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{(\Delta x)^2} \left\{ \left(\frac{(D_{i+1}^n + D_i^n)}{2}) \right) (u_{i+1}^n - u_i^n) - \left(\frac{(D_i^n + D_{i-1}^n)}{2} \right) (u_i^n - u_{i-1}^n) \right\} \tag{3.18}$$

After the integration from time t to $t + \tau$, the reaction terms of (2.6) and (2.7) are calculated and the losses subtracted from the density profile.

3.3.3 Fractional Step methods

Fractional step methods or Operator-Splitting methods, involve the decoupling of constituent processes with the appropriate numerical methods applied to each sub-problem sequentially. For the chemotaxis model studied by Tyson *et al.* [54], a high resolution explicit method was used for advection, and an implicit method for the diffusion. This model also included velocity fields that are not divergence free as the pattern formation intended to be produced depends on cells clustering in some regions. This is also applicable to soldier behaviour. It is noted that the interplay between pattern-erasing Fickian diffusion and pattern-creating chemotaxis creates observed stable patterns [54].

Fickian diffusion is analogous to the heat equation, that is, for a concentration ϕ, $\frac{\partial \phi}{\partial t} = D\nabla^2\phi$. Chemotaxis is a guided movement in response to a stimulus, generally applicable to cells or simple single/multicellular organisms, for example, the movement of bacteria in response to a chemical gradient.

In order to solve all terms simultaneously, the numerical method would typically need to be implicit due to diffusion. However, this would impose a severe time step restriction and lead to a method in which all terms are implicit and centred in space, potentially producing non-physical oscillations near discontinuities and steep gradients. Since these are features to be expected in this model, good resolution is critical. Many good methods exist that address each term's peculiarities well so a method which takes advantage of this is desired - fractional step methods.

Using a simple example of an advection-reaction equation to highlight the implications of the fractional step method: $q_t + \bar{u}q_x = -\beta q$ where the decay rate β does not depend on x, this can be split into two subproblems A: $q_t = -\bar{u}q_x$ and B: $q_t = -\beta q$. Solution operators for the two sub problems *commute* since solution methods can be applied in either order, flow then react or react then flow, producing the same result. Spatial dependency of β will introduce further error as the subproblems will then not be commutative producing results that will over or under-estimate actual results. Generally, some form of splitting error is introduced.

For the chemotaxis model which is a combination of reaction, diffusion and advection terms:

$$q_t = \mathcal{D}(q) + \mathcal{A}(q) + \mathcal{R}(q)$$

In order to obtain the solution q^{n+1} from q^n at time $t_{n+1} = t_n + \delta t$:

- Solve $q_t = \mathcal{A}(q)$ over time δt with data q^n to obtain q^*
- Solve $q_t = \mathcal{D}(q)$ over time δt with data q^* to obtain q^{**}
- Solve $q_t = \mathcal{R}(q)$ over time δt with data q^{**} to obtain q^{n+1}

Depending upon the precise forms of each of these terms, the appropriate numerical method for each step must be determined. Thus for our equations, we are able to use the methods described above for each of these terms giving first order accuracy.

- Solve $u_t = \mathbf{f_{diff}}$ over time δt with data u^n to obtain u^*
- Solve $u_t = \mathbf{f_{vel}}$ over time δt with data u^* to obtain u^{**}
- Solve $u_t = \mathbf{f_{react}}$ over time δt with data u^{**} to obtain u^{n+1}

Chapter 4

1D Results

We now apply the numerical method described in Chapter 3 to a one dimensional version of (2.6) with interaction terms removed. This is done in order to determine appropriate numerical parameters and compare results to previous swarming research by Mogilner *et al.*. The author has kindly provided the original Matlab code used to produce the published results. We expected to find good agreement of our numerical method with that used by Mogilner *et al.*. Both (2.6) and (2.7) including interaction terms are then modelled and compared to previous results of Protopopescu *et al.*. We expect to find similar reduction in density due to weapons effects however the density distributions should retain a relatively constant interior density with sharply defined edges throughout the simulations.

Once the numerical requirements have been established and we are satisfied that the interaction terms produce acceptable behaviour, this information will be used to direct the expansion of our equations to two dimensions.

4.1 Numerical Parameters and Previous Analytical Studies

Initially, we consider the force u in isolation. We take Equation (2.6), remove the firing terms $\mathbf{f_{react}}$ and write the equations in a non-linear conservation form. Equation (2.6) becomes:

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left((D\frac{\partial u}{\partial x}) + u(Cu + (A_a - A_r u)(K * u))\right) \tag{4.1}$$

with the kernel K defined as:

$$K(x) = \begin{cases} -(1/2r)sign(x) & -r \leq x \leq r \\ 0 & |x| > r \end{cases} \tag{4.2}$$

This is identical to the one dimensional system modelled by Mogilner *et al.*.

In [35], a linear stability analysis, analysis of the front and rear densities, and size and lifetime analysis was undertaken using a rectangular pulse ansatz solution and several key results found to ensure coherence of the formation. Firstly we highlight the findings for density independence.

- Constant diffusion will result in a small trail of swarm members being lost from the rear of the swarm.
- If the value of D is of the same order of magnitude or greater than:

  $$\frac{A_a r/4A_r}{(A_a - 4C)^2}, \tag{4.3}$$

  the density of stragglers behind the swarm will not be small resulting in the swarm dispersing in finite time.
- If

  $$D \ll \frac{A_a r/4A_r}{(A_a - 4C)^2} \tag{4.4}$$

  the layer of stragglers left will be exponentially thin.
- Loss of density will result in the interior density remaining the same while the width of the profile linearly decreases to compensate. Although the density loss in the analysis directly resulted from the drift of members away from the swarm when density is independent, this finding is expected to hold for a density loss due to firing effects in our simulations. The temporal dependence of the swarm width L is given by:

  $$\frac{dL}{dt} = \frac{CA_a}{A_r} exp\left[-\frac{A_a^2}{r} 4A_r D\left(1 - \frac{4C}{A_a}\right)\right] \tag{4.5}$$

The results for density dependence:

- Density dependent diffusion Df must be small enough,

$$D < \frac{A_a r/4}{(1-(2C/A_a))^2} \tag{4.6}$$

to ensure that the swarm remains stable and a true travelling band solution is found.

As we require a stable swarm with no loss of members, this small density dependent diffusion is essential and our parameters much be chosen to satisfy the stability conditions. We follow Mogilner *et al.* by choosing the density dependent diffusion to be of the form:

$$D = \begin{cases} D_u u & \text{if } u < 0.1 \times A_a/A_r \\ D_u & \text{otherwise,} \end{cases} \tag{4.7}$$

where D_u is a constant.

4.1.1 Flux Limiters, Tolerances and Grid Spacing

In order to determine an accurate solution to be found within an acceptable computational time, the choice of flux limiter, tolerance and grid spacing is of the utmost importance. Not only is capturing steep fronts formed by each force essential, the steepness of the rear of the profile must also be approximated well. This can be achieved if the limiter is of high enough accuracy yet prevents numerical oscillations or negative densities. Several first and second order standard and well known flux limiters as given in Table 4.1 were investigated for a range of tolerances and compared to a high tolerance second order van Leer limiter result.

Typically for simulations of this nature, time is stated in terms of the number of time steps taken. As this is normal for fixed sized time steps and the numerical method used here relies on a variable time step, we leave time values in a less familiar format of, in this case, decimals. For Figure 4.2 an end time of $t = 0.001$ is used such that the force moves approximately one third of the length of the domain.

Name	Limiter
First Order Upwind	0
van Leer	$\frac{r+\|r\|}{1+\|r\|}$
Second Order Central	$max\left\{0, min(r, \delta)\right\}$
Second Order Upwind	$max\left\{0, min(2r, \delta, 1)\right\}$

Table 4.1: Potential Flux Limiters.

Following Boswell, the parameter δ for the Second Order Central and Second Order Upwind limiters was set to 2.

$log_{10}(tol)$	Flux Limiter							
	First Order Upwind		van Leer		Second Order Central Difference		Second Order Upwind	
	Time	RS	Time	RS	Time	RS	Time	RS
-2	0.703	0	0.782	0	0.563	0	0.766	0
-3	2.109	1	2.407	1	2.531	5	2.407	1
-4	6.562	3	6.906	24	8.078	30	6.578	25
-5	19.469	2	20.5	138	23.031	145	20.438	109
-6	62.203	3	60.187	211	75.266	242	64.312	194
-7	195.859	5	186.344	273	233.218	302	210.8248	249

Table 4.2: Run Time and Rejected Steps (RS) for selected Limiters.

For Figure 4.2, the initial density profile is:

$$u(x,0) = \begin{cases} 10 & 0.1 \leq x \leq 0.3 \\ 0 & \text{otherwise} \end{cases}$$

Parameters are $r = 5, D = 5, C = 20, A_a = 20, A_r = 2, \Delta t = 10^{-6}$, end time $= 10^{-3}$.

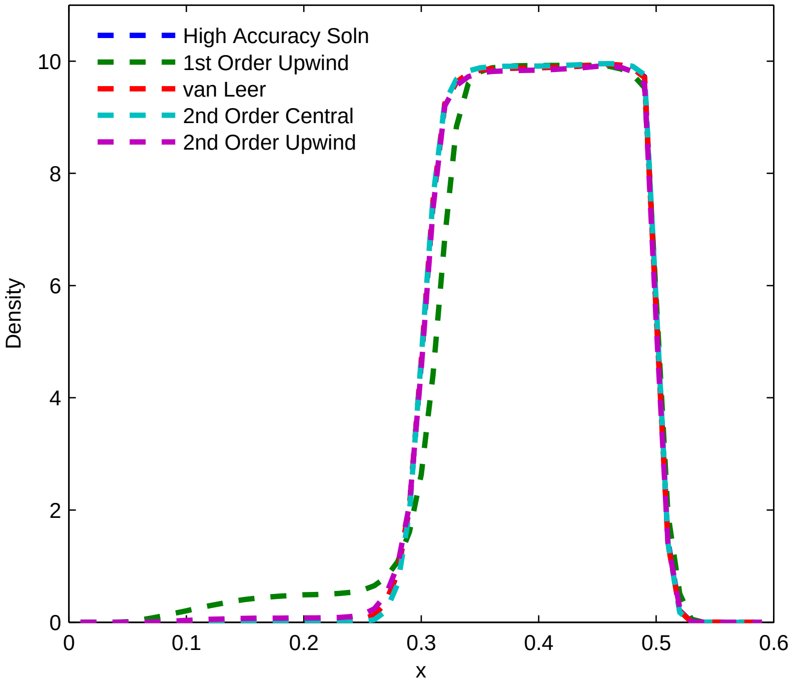


Figure 4.1: All limiters have a tolerance of 10^{-7} and the high accuracy solution using the van Leer limiter with a tolerance of 10^{-10}

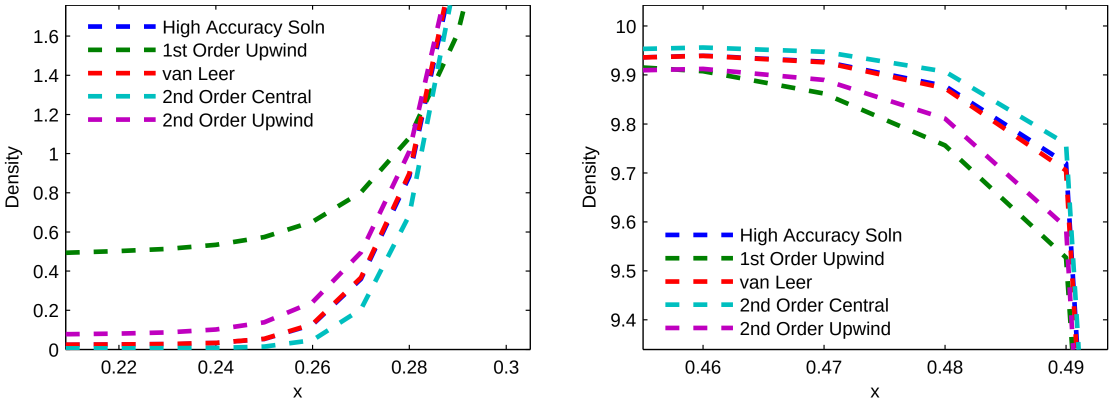


Figure 4.2: Comparison of Tolerances

All results were generated using a HP Compaq DC7800 Intel Core 2 Duo CPU 2.66GHz 1.95GB RAM personal computer.

For each Rejected Step entry in Table 4.1, the majority of the rejected steps occurred during the initial start up phase. As the initial time step $\Delta t = 10^{-6}$ produced a larger error ρ for higher tolerances, a higher number of time step corrections were required for the error to fall within acceptable limits as detailed in

Section 3.16. As each correction increments the rejected steps value, it follows that a higher tolerance will produce more rejected steps in reaching an adequate error.

While both the first and second order upwind limiters gave results in the same order of magnitude of computational time and rejected steps, their inability to resolve the rear of the force such that a trailing profile is left, removes them from consideration as potential limiters for this model. The First Order Upwind gives the least accuracy with poor resolution especially of the rear of the profile. Both the van Leer and Second Order Central limiters are in close agreement with the high accuracy solution. As the van Leer limiter is continuously differentiable and has good positivity properties at lower tolerances [3], this will be the limiter used for the one dimensional results presented in this chapter.

Having established a tolerance, the effect of grid spacing on the solution is now investigated. Note that the computational times given in Table 4.3 are for one force profile only whereas we will be simulating the interaction of two forces and their interaction. Parameters for Figure 4.3 are $D = 5, C = 20, A_a = 20, A_r = 2, r_{a,r} = 5, \tau(t = 0) = 10^{-7}, atol = rtol = 10^{-6}$, end time $t = 10^{-3}$.

Δx	$log_{10}(tol)$									
	-3		-4		-5		-6		-7	
	RS	Time	RS	Time	RS	Time	RS	Time	RS	Time
0.01	1	0.7	23	2	142	6.36	214	19.8	274	61.4
0.005	2	3.2	2	10.1	270	32.9	760	105.7	1044	326.6
0.0025	3	32.9	3	104.7	7	316	1560	1002.2	-	-
0.00125	4	1517.1	5	4859.3	-	-	-	-	-	-

Table 4.3: Comparison of Grid Spacings and Tolerances to Computational Time and Rejected Steps.

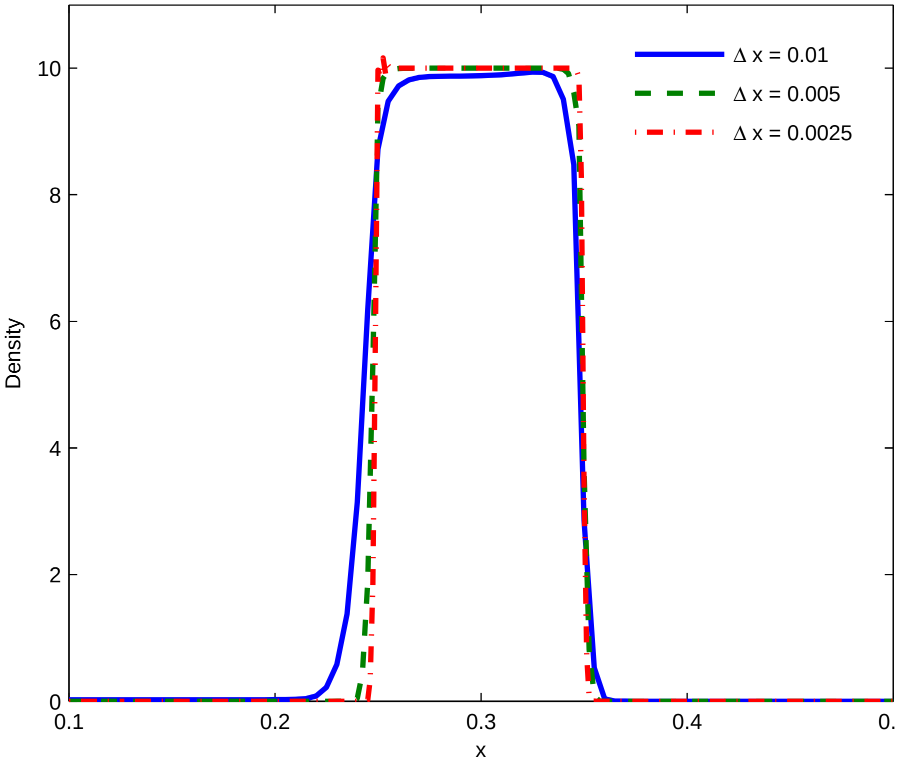


Figure 4.3: Comparison of Grid Spacings

For our equations, the Péclet number which indicates the degree to which advection dominates diffusion is of the order:

$$P = \frac{C\Delta x}{D} = \frac{60 * 0.005}{10} = 0.03. \tag{4.8}$$

This indicates that advection does not dominate diffusion and that spurious oscillations associated with applying finite difference schemes to high Péclet numbers does not apply in this case.

As can be seen in Figure 4.3, the location of the leading and trailing edge of the force profile does not greatly alter between $\Delta x = 0.005$ and 0.0025. In addition, the computational time increases greatly between these two values and as such, the grid spacing and tolerance values to be used for the one dimensional modelling will be 0.005 and 10^{-6} respectively.

4.1.2 Comparison with Previous Work

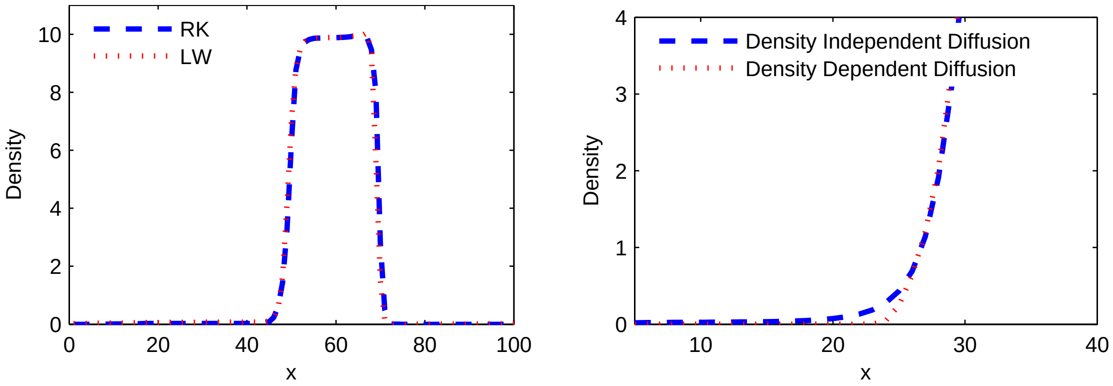
Now that the tolerance and limiter have been determined, we compare results against the original algorithm kindly provided by Alex Mogilner in order to test the numerical method of integration proposed. The ranges at which attraction and repulsion operate are taken to be equal ($r = 5$). Diffusion is set at ($D = 5$) and convection ($C = 20$). Convolutions were approximated using a second order accurate trapezoidal rule. Unless otherwise stated, the domain was discretised into 100 grid points. For Figure 4.4a the initial density profile is:

$$u(x,0) = \begin{cases} 10 & 0.1 \leq x \leq 0.3 \\ 0 & \text{otherwise} \end{cases}$$

Parameters for Mogilner's results are $r = 5, D = 5, C = 20, A_a = 20, A_r = 2, \Delta t = 10^{-6}$, number of time steps = 2000. For our numerical method, all parameters are identical with the exception of $\tau(t = 0) = 10^{-6}, atol = rtol = 10^{-6}$. For Figure 4.4b,

$$D = \begin{cases} D_u u & \text{if } u < 0.1 \times A_a/A_r \\ D_u & \text{otherwise} \end{cases}$$

as specified for density dependent diffusion with $D_u = 5, A_a = 20, A_r = 2, r = 5, \tau(t = 0) = 10^{-6}, atol = rtol = 10^{-6}$, end time $t = 10^{-3}$. The initial density profile is the same as in Figure 4.4a.



(a) Comparison of the numerical method used by Mogilner (Lax-Wendroff) and that proposed in Section 3 (Runge-Kutta).

(b) A comparison of density independent ($D = D_u$) and density dependent diffusion.

Figure 4.4: Comparisons with Mogilner *et al.'s* Numerical Method.

Figure 4.4a demonstrates a good agreement with Mogilner *et. al*'s results and the troop formation holds. Oscillations present in Mogilner's results, more easily seen in videos or shorter interval snapshot series, are reduced due to the flux limiter and the density of lost organisms behind the troop profile is slightly higher. Had the advection term not included the convolution terms, we would expect the density profile to retain a similar shape with diffusive spread increasing throughout the simulation. This diffusive spread is easily seen in the results of Protopopescu where the initial Gaussian profile slowly diffuses throughout the simulation.

Diffusion is now taken as density dependent Du for densities below $u < 0.1 \times (A_a/A_r)$ and constant D otherwise. Figure 4.4b shows a zoom of the rear of the profile highlighting the noticeable difference between density dependence and independence. As expected, there is no lost density behind the profile when diffusion is density dependent. This also agrees with Mogilner *et al.*'s findings in one dimension.

As the behaviour of the moving troop profile is satisfactory, we proceed to introduce the firing terms. Following military and wargaming nomenclature, forces are named Red and Blue.

4.2 Force Interaction

We now consider interaction between two forces by reintroducing the firing terms $\mathbf{f}_{\mathbf{react}}$. Firstly, the simpler aimed fire term is included so that soldier losses are sustained only when troop profiles overlap in the domain. Secondly, only the area fire term is included resulting in spatially dependent losses. Thirdly, both aimed and are fire terms are included and lastly, these results are compared to Protopopescu's results.

4.2.1 Aimed Fire Only

No changes in tactics due to losses incurred are implemented, e.g. retreat upon a 50% reduction. Each force maintains a constant velocity regardless of losses and will pass directly through the opposing force. Figure 4.6 demonstates that losses only occur when members of the opposing forces are co-located as expected. The initial density profile is:

$$u(x,0) = \begin{cases} 10 & 0.1 \leq x \leq 0.3 \\ 0 & \text{otherwise} \end{cases}, \quad v(x,0) = \begin{cases} 10 & 0.7 \leq x \leq 0.9 \\ 0 & \text{otherwise} \end{cases}$$

Parameters were set at $r_{u,v} = 5, D_{u,v} = 5, C_u = 20, C_v = -20 A_{a_{u,v}} = 20, A_{r_{u,v}} = 2, d_{u,v} = 8 \times 10^{-5}, \tau(t=0) = 10^{-6}$, end time $t = 2 \times 10^{-3}, atol = rtol = 10^{-6}, \Delta x = 0.005$.

$$\frac{\partial u}{\partial t} = \nabla \cdot (\mathbf{D_u}(u) \nabla u) + \nabla \cdot \{u(\mathbf{C_u} u + (A_a - A_r u)(K * u))\} + d_u v \tag{4.9}$$

$$\frac{\partial v}{\partial t} = \nabla \cdot (\mathbf{D_v}(v) \nabla v + \nabla \cdot \{v(\mathbf{C_v} v + (A_a - A_r v)(K * v))\} + d_v u \tag{4.10}$$

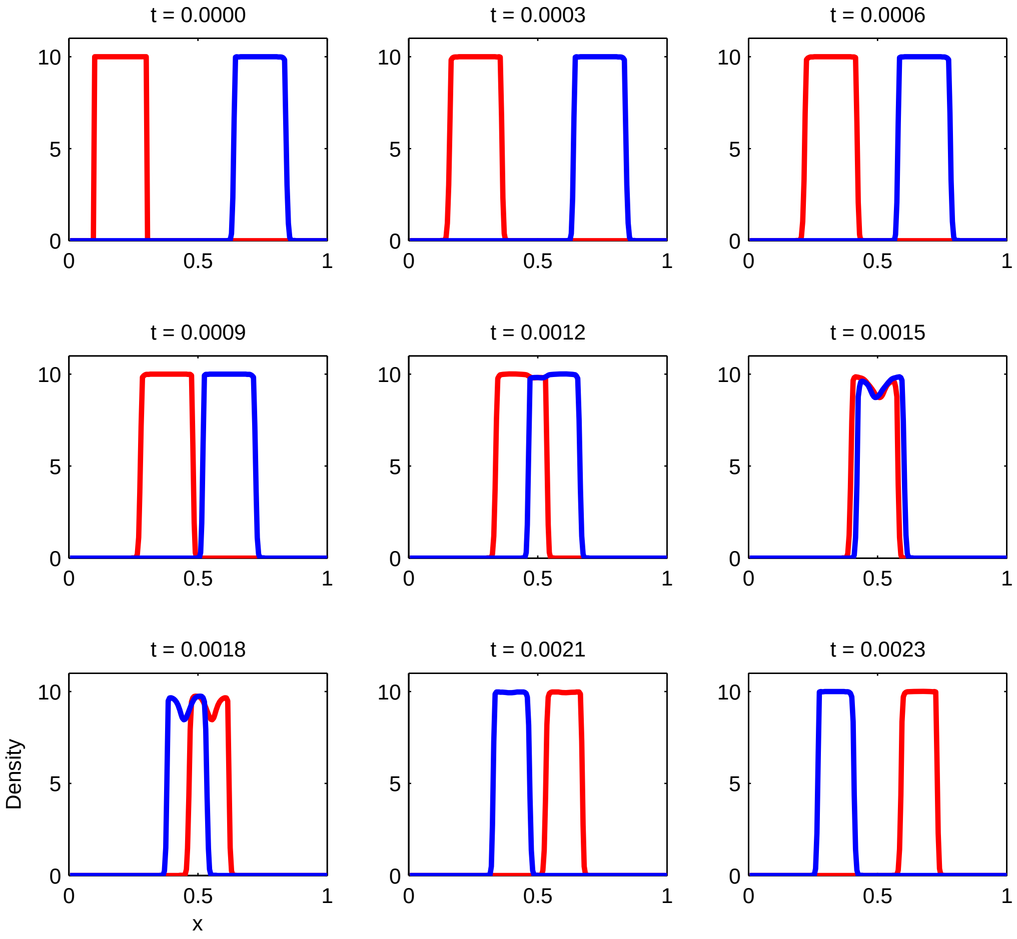


Figure 4.5: 1D Aimed Fire Only

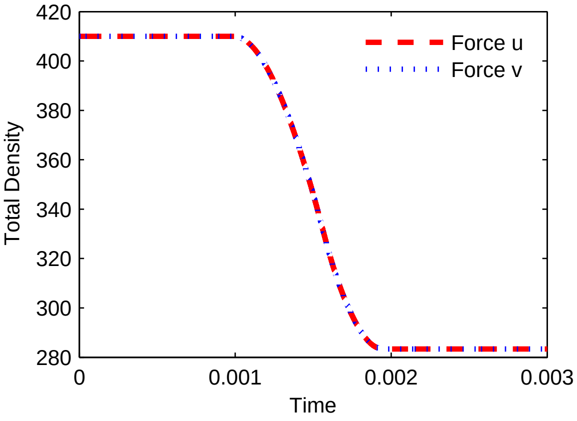


Figure 4.6: 1D Aimed Fire Only Losses

As can be seen in Figure 4.6 troops are rapidly killed when they come in contact due to the high value of $d_u = d_v = 8 \times 10^{-5}$ and the constant velocity sufficiently low that the two forces have enough time to inflict significant casualties. At $t = 8 \times 10^{-4}$, where approximately half of each force is in contact with each other, the drop in density due to fire can be clearly seen. As the forces pass each other and incur more losses, the densities fall below the ratio $A_a/A_r = 10$. The effects of the attraction kernel becomes apparent in Figure 4.5 as each force contracts in order to maintain a constant interior density of 10 with steep edges throughout the simulation. At $t = 1.8 \times 10^{-3}$, there is no overlap of forces and thus no firing effects resulting loss of density. Each force now exhibits a stable travelling profile that is thinner than the initial profile due to reduced numbers.

4.2.2 Area Fire Only

Of the f_{react} terms, we now consider only the Area Fire terms:

$$\frac{\partial u}{\partial t} = \nabla \cdot (\mathbf{D_u}(u)\nabla u) + \nabla \cdot \{u(\mathbf{C_u}u + (A_a - A_r u)(K * u))\} + u\,(k_u * v) \qquad (4.11)$$

$$\frac{\partial v}{\partial t} = \nabla \cdot (\mathbf{D_v}(v)\nabla v) + \nabla \cdot \{v(\mathbf{C_v}v + (A_a - A_r v)(K * v))\} + v\,(k_v * u) \qquad (4.12)$$

where the form of the Area Fire kernel is:

$$k(x) = \beta e^{-\nu|(x-X)|}, \qquad (4.13)$$

and is taken to act over the whole domain.

As the form of area fire is proportional to the density, initial losses are expected to be evenly felt across the troop formation. As the initial density was set to the desired A_a/A_r, any loss incurred will result in the density dropping below this value and the attraction term dominating over the repulsion term. This forces the troops to contract so as to maintain this desired density, unlike traditional advection/diffusion/reaction models. This is seen clearly in Figure 4.7.

The initial density profile is:

$$u(x,0) = \begin{cases} 10 & 0.1 \le x \le 0.3 \\ 0 & \text{otherwise} \end{cases}, \quad v(x,0) = \begin{cases} 10 & 0.7 \le x \le 0.9 \\ 0 & \text{otherwise} \end{cases}$$

Parameters were set at $r_{u,v} = 5, D_{u,v} = 5, C_u = 20, C_v = -20A_{a_{u,v}} = 20, A_{r_{u,v}} = 2, \beta_{u,v} = 8 \times 10^{-5}, \nu_{u,v} = 0.8, \tau(t=0) = 10^{-6}$, end time $t = 2 \times 10^{-3}$, $atol = rtol = 10^{-6}, \Delta x = 0.005$.

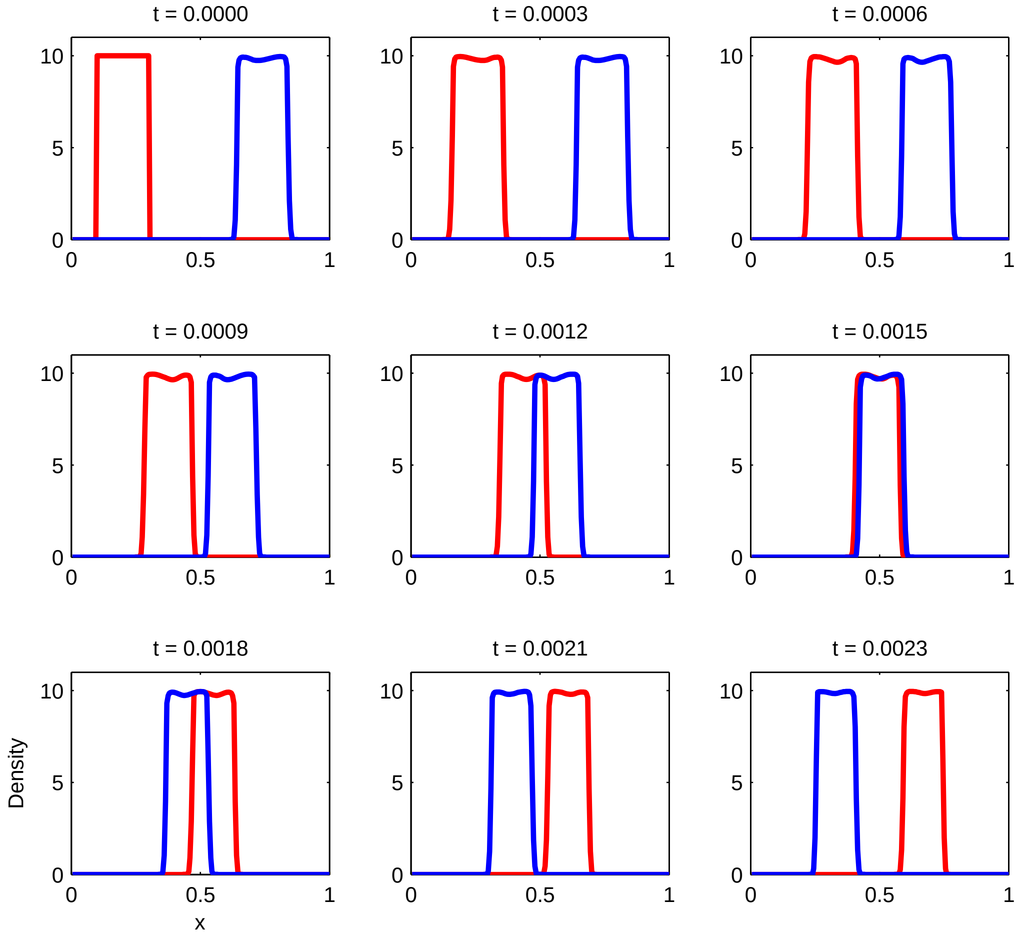


Figure 4.7: 1D Area Fire Only

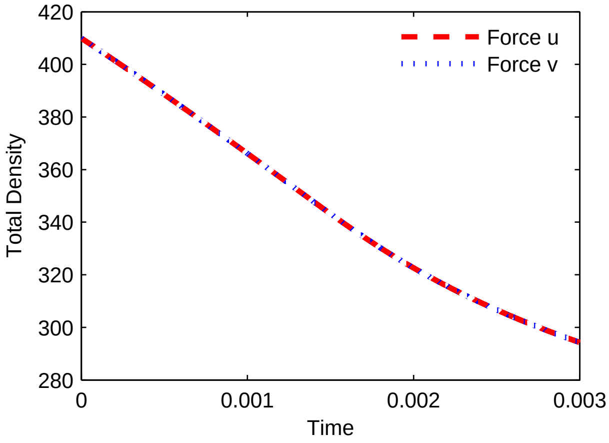


Figure 4.8: 1D Area Fire Only Losses

4.2.3 Both Aimed and Area Fire

We now include both Aimed and Area Fire, giving (2.6) and (2.7). We see an immediate reduction in both forces due to the non-local effects of Area Fire, and an increased reduction due to Aimed Fire effects when the forces come into contact. The initial density profile is:

$$u(x,0) = \begin{cases} 10 & 0.1 \leq x \leq 0.3 \\ 0 & \text{otherwise} \end{cases}, \quad v(x,0) = \begin{cases} 10 & 0.7 \leq x \leq 0.9 \\ 0 & \text{otherwise} \end{cases}$$

Parameters were set at $r_{u,v} = 5, D_{u,v} = 5, C_u = 20, C_v = -20, A_{a_{u,v}} = 20, A_{r_{u,v}} = 2, \beta_{u,v} = 8\times 10^{-5}, \nu_{u,v} = 0.8, d_{u,v} = 8\times 10^{-5}, \tau(t=0) = 10^{-6}$, end time $t = 2\times 10^{-3}$, $atol = rtol = 10^{-6}, \Delta x = 0.005$.

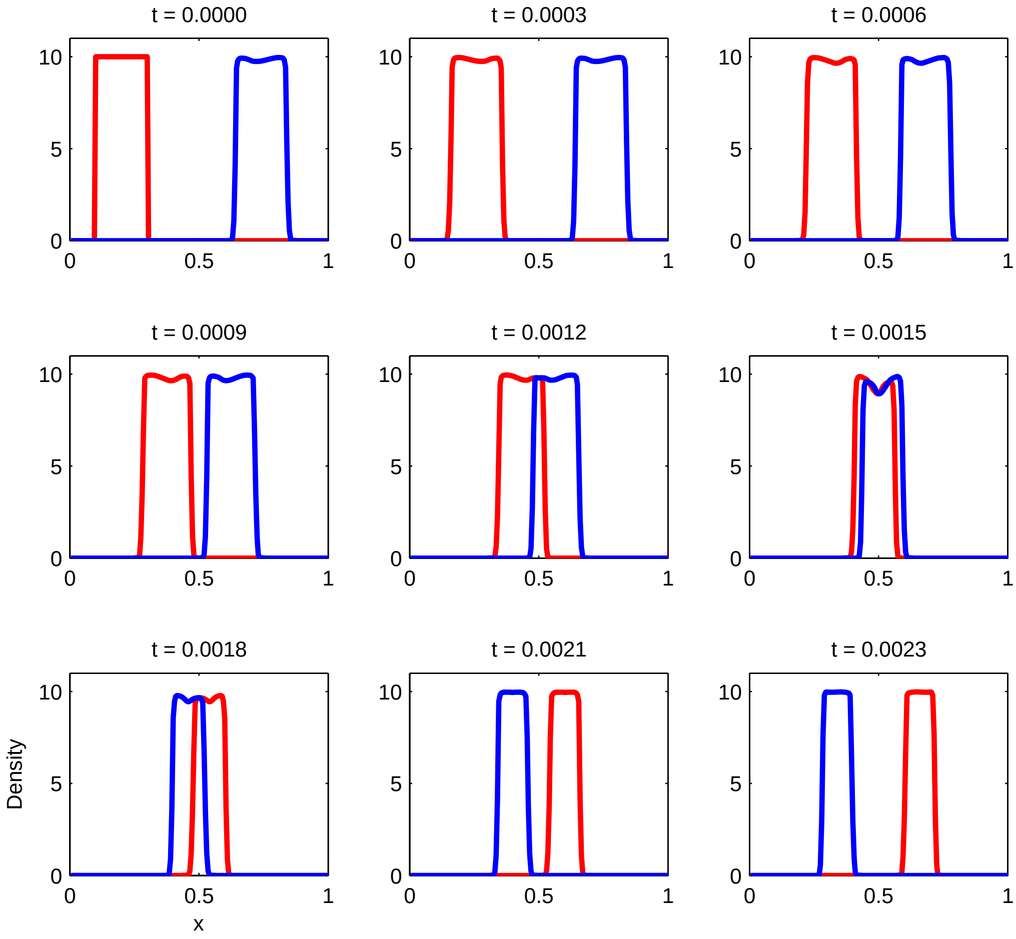


Figure 4.9: 1D Aimed and Area Fire Only.

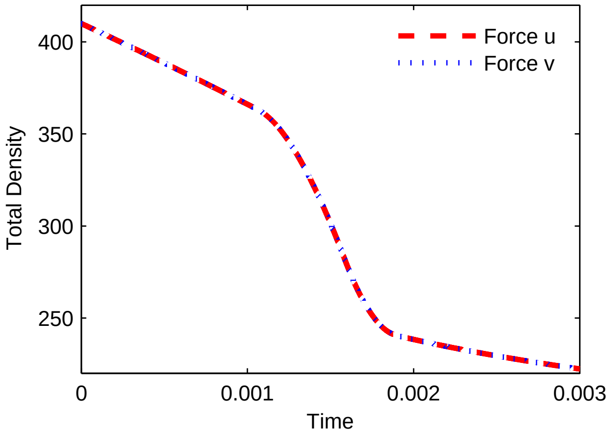


Figure 4.10: 1D Aimed and Area Fire Losses

4.2.3.1 Comparison to Previous Research

Protopopescu *et al* generally enforced higher level decisions or tactics into their simulations which we have chosen to forego. Therefore, comparisons are restricted due to the small number of published results available that do not include these tactics. However in [46], Figure 4 is comparable to Figure 4.7 and shows a similar trend in soldier losses. The parameters used for Figure 4.11 are $L = 5, C_u = 5, C_v = -6, D_u = 0.01, D_v = 0.02, c_u = -2.4 \times 10^{-4}, c_v = -3 \times 10^{-4}, \nu_{u,v} = 1/3, u(t=0) = v(t=0) = 2000$:

$$\frac{\partial u}{\partial t} = D_u \frac{\partial u}{\partial x^2} + C_u \frac{\partial u}{\partial x} - c_u u \int_{-L}^{L} e^{-\nu_u |x-y|} v(y) dy \tag{4.14}$$

$$\frac{\partial v}{\partial t} = D_v \frac{\partial v}{\partial x^2} + C_v \frac{\partial v}{\partial x} - c_v v \int_{-L}^{L} e^{-\nu_v |x-y|} u(y) dy \tag{4.15}$$

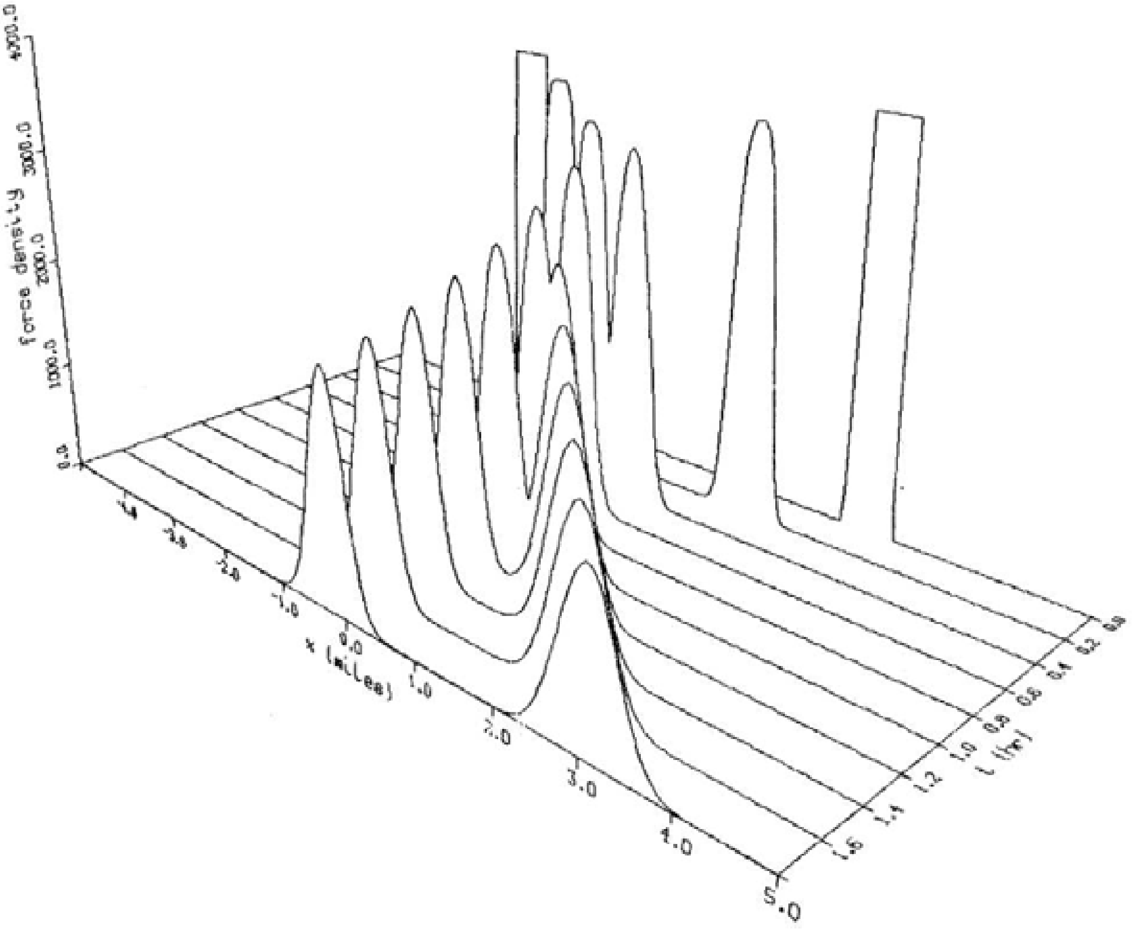


Figure 4.11: Protopopescu *et al*'s results for Aimed Fire. Reproduced from [46].

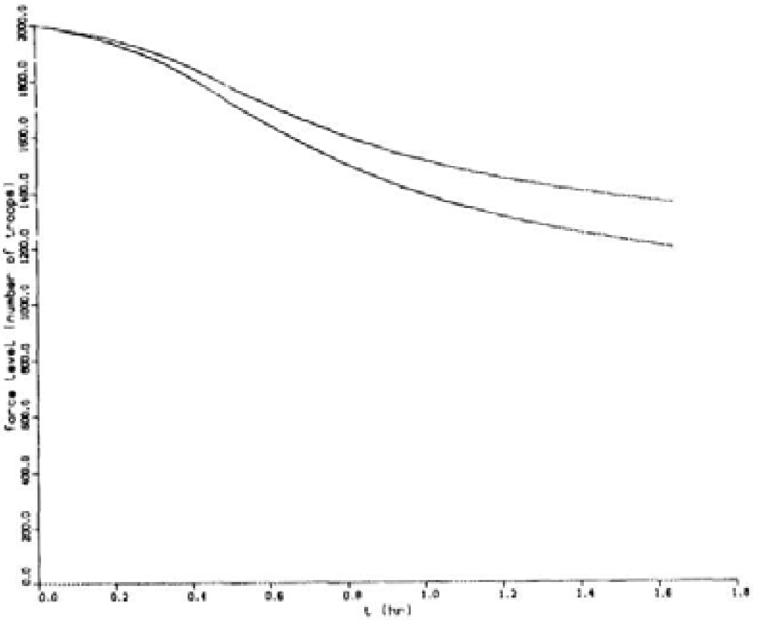


Figure 4.12: Losses for Figure 4.11. Reproduced from [46].

Figure 4.11 shows obvious smearing of the force profile with time and thus the complete absence of the ability to maintain a constant interior density with well defined edges. Density loss as a function of time is also of the same form as our Area Fire Only results indicating the correct $\mathbf{f_{react}}$ form as the essence of Protopopescu's results have been captured. Therefore, even our simplest one dimensional model with a basic interaction term is able to model soldier movement with a greater degree of realism with regards to spatial movement.

Now we can extend these equations to two dimensions in preparation for comparisons to the cellular automaton wargame ISAAC.

Chapter 5

2D Results

We now extend Equations (2.6) and (2.7) to two dimensions. Initially we determine the appropriate tolerance to be used in a similar manner as for the one dimensional case, and the appropriate method for solving the convolution terms in a computationally acceptable time. We then proceed to investigate the interactions between two forces in a comparable manner as Chapter 4. Lastly we look at the introduction of obstacles in the domain and the navigation of a force around these obstacles.

5.1 Specifics of the Numerical Method

5.1.1 Numerical Methods for Convolution terms

Moving from one to two dimensions presented significant concerns with regards to computational time required. This is due to a total of three computationally expensive convolution terms for each force - attraction, repulsion and area fire - that must be calculated for each of the $N \times N$ grid points for each time step. Two approaches available in the standard version of Matlab 2008b for solving convolutions were used. Firstly, the standard convolution functions **conv** and **conv2** were utilised and a significant speed increase was noticed. Secondly, Fourier theory for convolutions was implemented using Matlab's Fast Fourier (FFT) and Inverse Fast Fourier Transformation (IFFT) functions **fft2** and **ifft2**. Using standard Fourier transform theory, Fourier transformations of both functions to be convoluted were taken, multiplied together, and the inverse fast Fourier transform applied.

$$\frac{1}{2\pi}\int_{-\infty}^{\infty}\hat{f}(\alpha)\hat{g}(\alpha)e^{i\alpha x}d\alpha = \int_{-\infty}^{\infty}f(x-y)g(y)dy$$

Using the parameters $D = 2, \boldsymbol{C} = (20, 20), A_a = 20, A_r = 2, r_{a,r} = 2, \tau(t = 0) = 10^{-7}, atol = rtol = 10^{-4}$ and $\Delta x = 0.01$, the times taken to reach $t = 0.001$ are summarized in Table 5.1.

	fft2, **ifft2**	**conv2**
Run Time	358.547	116.484
	351.797	123.422
	360.157	116.187
	358.11	114.062
	353.156	116.235
	349.454	112.265
	356.75	118.031
	356.829	119.61
	351.015	116.906
	355.954	113.735
Mean	355.1769	116.6937
Standard Deviation	3.59	3.19

Table 5.1: Run Times for Convolution Comparisons.

Noting the increase in time using the FFT/IFFT approach for the given grid size, the convolution function **conv2** will be used for the following numerical simulations. While using larger grid sizes results in time savings for the FFT/IFFT approach, as the maximum value of N being considered is 100, the **conv2** approach is sufficient for this research.

5.1.2 Variation of Tolerances and Grid Spacing

Following the one dimensional equations, in order to determine an accurate solution to be found within acceptable computational time, a comparison of tolerances to computational time was undertaken. In order for the simulation time to fall within acceptable limits, the tolerance was varied between 10^{-2} and 10^{-5}. As for the one dimensional results, the van Leer limiter is used throughout the two dimensional results. The initial density profile is:

$$u(x,y,0) = \begin{cases} 10 & \sqrt{5/(5 \times \sqrt{2\pi})} \times e^{-((x-0.15)^2+(y-0.15)^2)/(0.5)} > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

Parameters are $D = 10, \boldsymbol{C} = (60, 60), A_a = 5, A_r = 0.5, r_{a,r} = 5, \tau(t = 0) = 10^{-7}$, end time $t = 0.001$ and $\Delta x = 0.01$.

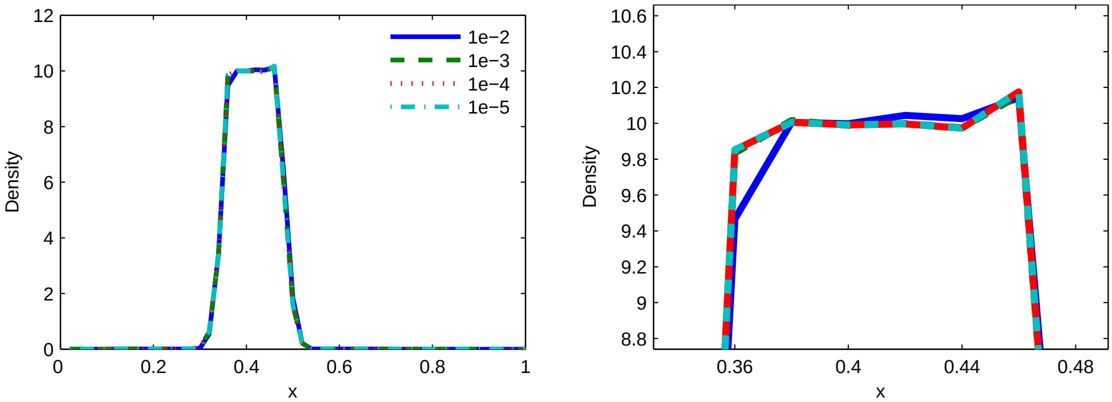


Figure 5.1: Comparison of tolerance values, horizontal slice of the 2D profile.

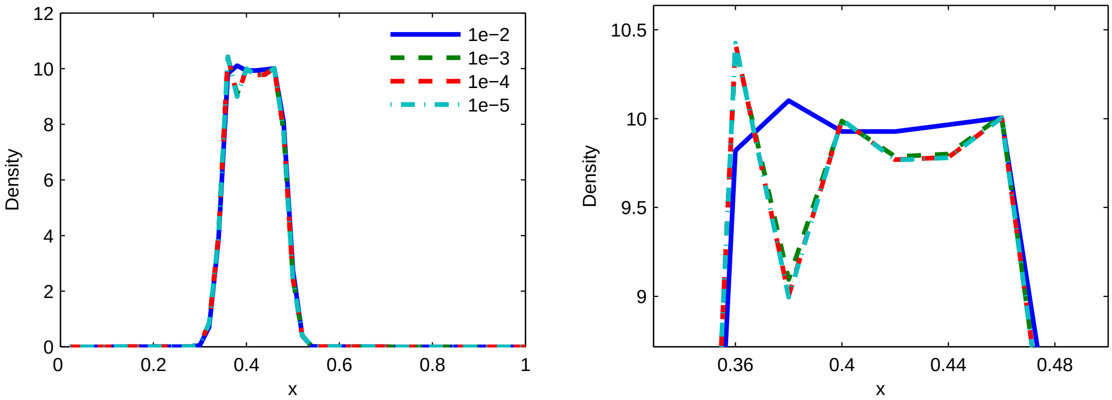


Figure 5.2: Parameters are identical to Figure 5.1.

Tolerance	Rejected Steps
$1e^{-2}$	0
$1e^{-3}$	1
$1e^{-4}$	4
$1e^{-5}$	6

Table 5.2: Rejected Steps for Tolerances in 2D.

The tolerance value of 10^{-3} gives an acceptable level of accuracy within a reasonable computational time and thus will be used for all two dimensional results presented. For all tolerance values rejected steps occurred only at the initial stages of the simulation and once an acceptable value of τ was found, no further rejected steps occurred. We use a tolerance value of 10^{-5} as the control high accuracy result.

Having established a tolerance and in a similar manner as for the one dimensional results, the effect of grid spacing on the solution is now investigated. Note that the computational times given in Table 5.3 are for one force profile only whereas we will be simulating the interaction of two forces and their interaction. Parameters for Figure 5.3 are $D = 5, C_x = 60, C_y = 0, A_{a_{x,y}} = 5, A_{r_{x,y}} = 0.5, r_{a,r_{x,y}} = 5, \tau(t = 0) = 10^{-7}, atol = rtol = 10^{-3}$, end time $t = 10^{-4}$.

Δx	Rej Steps	Time
0.02	0	10.3590
0.01	0	92.7460
0.005	1	460.4520
0.0025	3	2228.9

Table 5.3: Comparison of Grid Spacings.

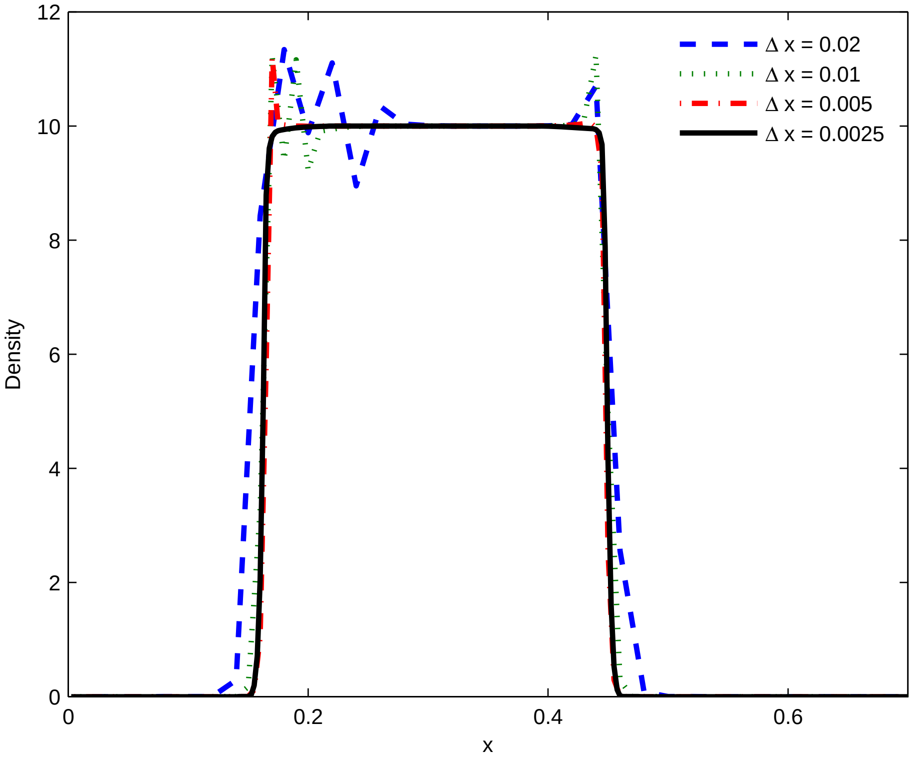


Figure 5.3: Comparison of Grid Spacings

For our equations, the Péclet number for the interior of the profile is:

$$P = \frac{C\Delta x}{D} = \frac{60 * 0.01}{5} = 0.03. \tag{5.1}$$

This indicates that advection does not dominate diffusion and that spurious oscillations associated with applying finite difference schemes to high Péclet numbers does not apply in this case.

As can be seen in Figure 5.3, the location of the leading and trailing edge of the force profile does not greatly alter between $\Delta x = 0.01$ and 0.0025 however the best overall density profile resolution is found with grid spacing 0.0025. In addition, the computational time increases greatly between these two values and as such, the grid spacing and tolerance values to be used for the one dimensional modelling will be 0.01 and 10^{-3} respectively.

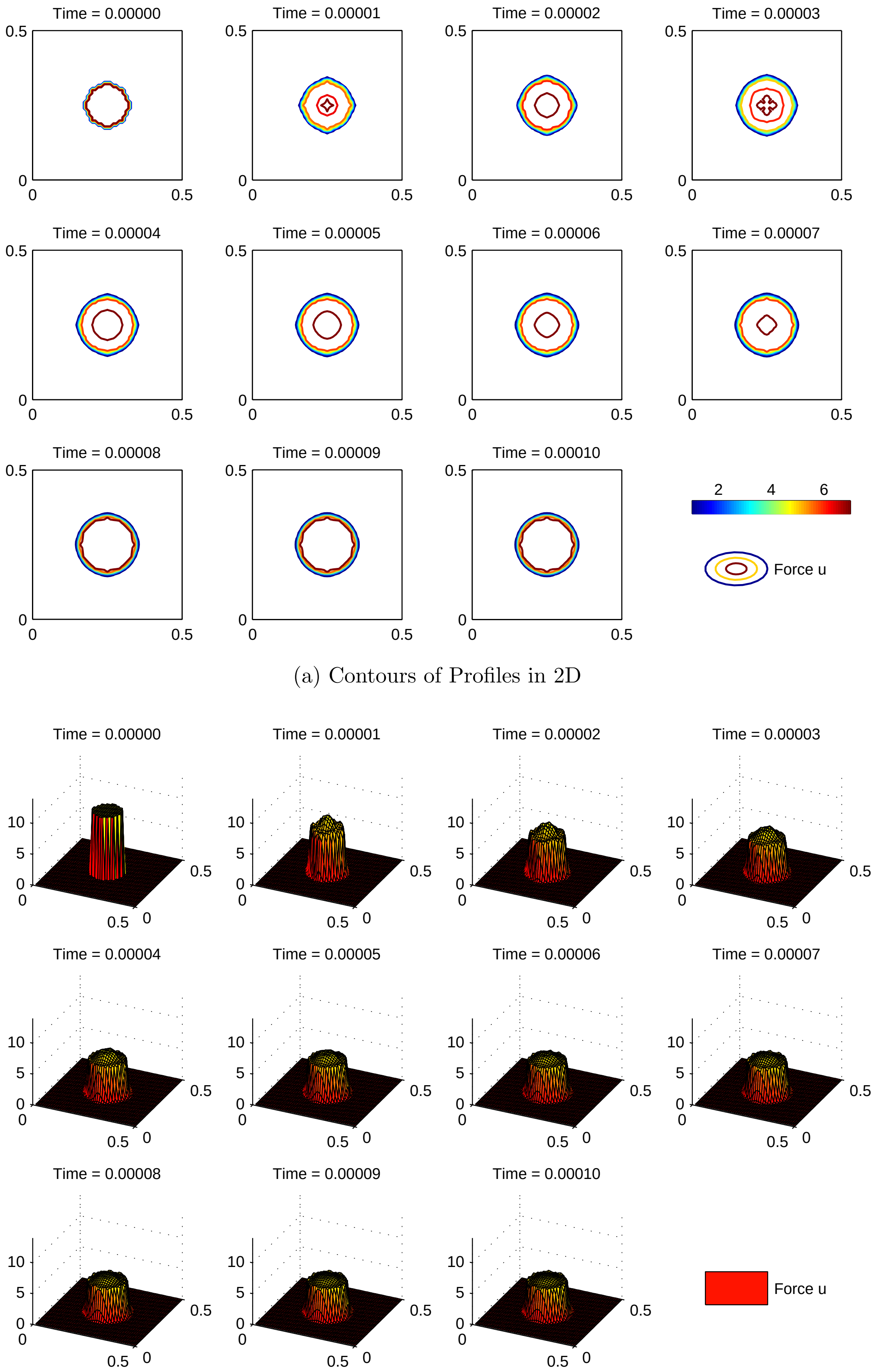
5.2 Movement Only

5.2.1 Effect of Attraction and Repulsion

Considering one force in isolation (no firing effects), setting the convection term to zero and using density dependent diffusion, the asymptotic behaviour is investigated. As for the one dimensional version, it is expected that the resulting profile will be circular and will retain a constant interior density with steep edges dropping to zero. We set the initial profile as non-circular and confirm that the asymptotic stationary state is as a circular profile with constant interior density of A_a/A_r and steep circumference with density falling to zero. The initial density profile is:

$$u(x, y, 0) = \begin{cases} 10 & \sqrt{(x-0.15)^2 + (y-0.15)^2} < 0.071 \\ 0 & \text{otherwise} \end{cases}$$

Parameters were set at $r_{a,r} = 5, D = 10, \boldsymbol{C} = 0, A_a = 5, A_r = 1, \tau(t = 0) = 10^{-8}$, end time $t = 10^{-2}$, $atol = rtol = 10^{-4}, \Delta x = 0.01$.



(a) Contours of Profiles in 2D

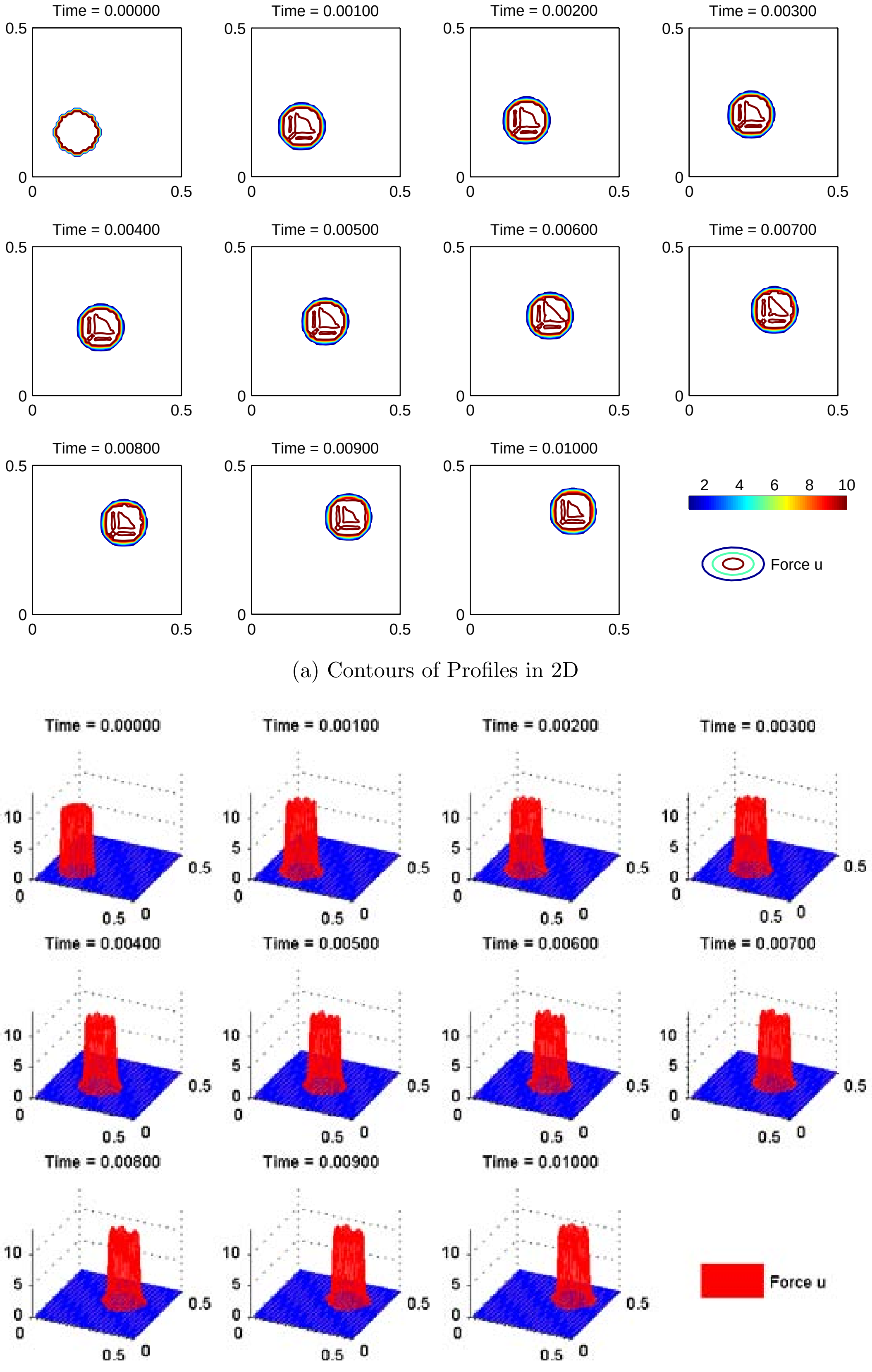
(b) Graphs of profiles 3D

Figure 5.4: Asymptotic State for Stationary Force Profile

We now set the velocity to a constant value in order to ascertain whether the resolution of the steep moving fronts is sufficient. That is, whether the movement throughout the battlespace is realistic. An initial circular troop profile begins in the lower left quadrant of the domain and proceeds with constant velocity toward the upper right corner of the domain. Snapshots are taken at approximately equal time intervals. The initial density profile is:

$$u(x, y, 0) = \begin{cases} 10 & \sqrt{(x-0.15)^2 + (y-0.15)^2} < 0.071 \\ 0 & \text{otherwise} \end{cases}$$

Parameters were set at $r_{a,r} = 5, D = 10, \boldsymbol{C} = (60, 60), A_a = 20, A_r = 2, \tau(t = 0) = 10^{-7}$, end time $t = 10^{-2}$, $atol = rtol = 10^{-3}, \Delta x = 0.01$.



(a) Contours of Profiles in 2D

(b) Graphs of profiles 3D

Figure 5.5: Diagonal Movement Only

As Figure 5.5 demonstrates, the troops move in a cohesive fashion, approximately maintaining the initial profile of constant interior density with steep edges. This behaviour is a noticeable feature of published ISAAC results [18], and is labelled as "*unexpected self-organized internal formation*". However, we have shown that such self-organisation may be modelled using relatively simple and physically meaningful PDEs that are comprehensible enough to allow us to predict this behaviour *a priori*. For the cellular automata results though, this was unexpected and required simulations to be run.

5.3 Inclusion of Interaction Terms

As with the one dimensional results, aimed and area fire are initially treated separately and then combined.

5.3.1 Aimed Fire Only

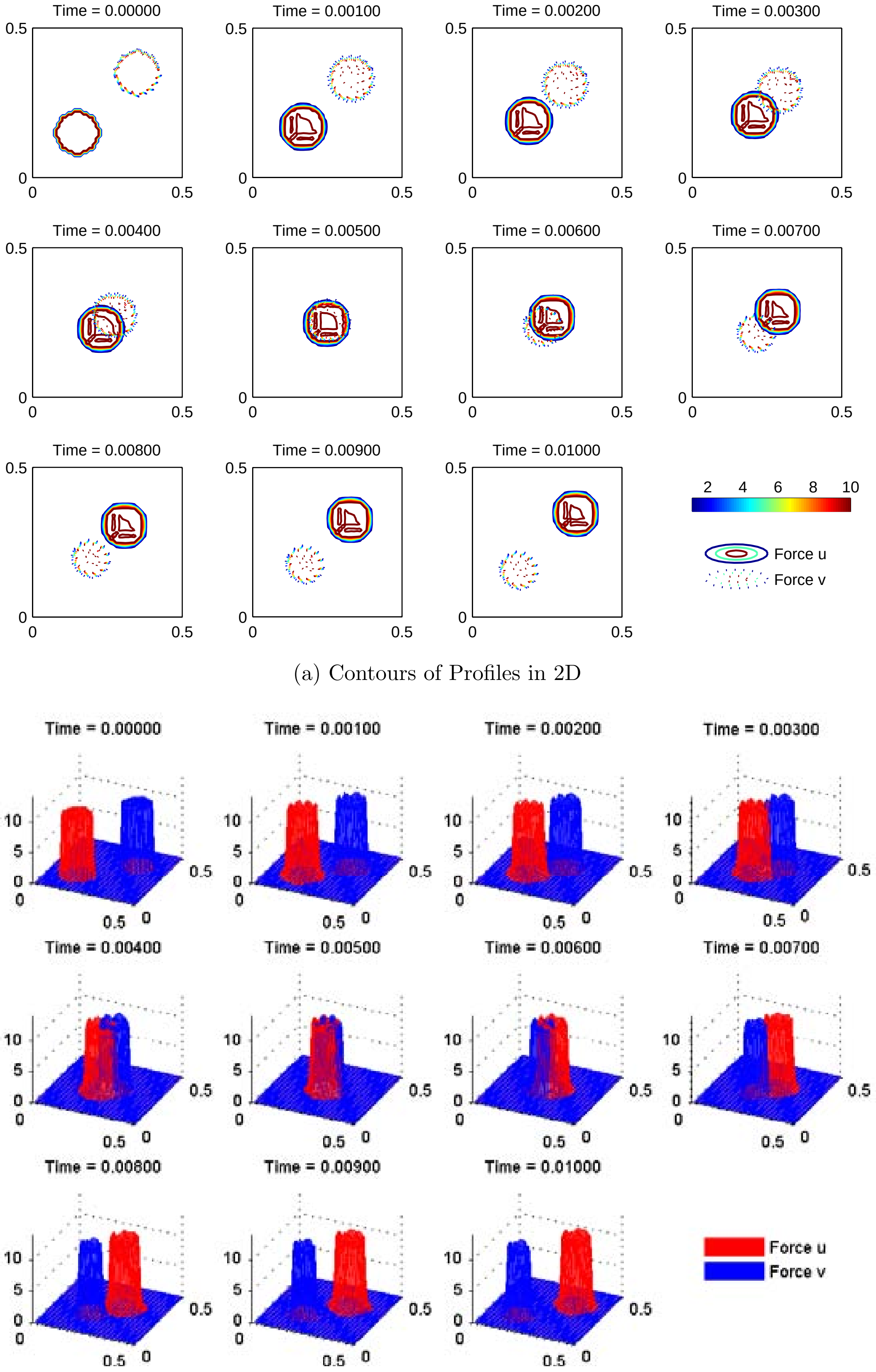
We now investigate the aimed fire terms only. An opposing force is introduced, and the effect of area and aimed firing terms determined. Initial circular profiles with interior densities of A_a/A_r in opposite quadrants are set, contour intervals of the Matlab graphs are fixed at equal intervals with one force represented as dashed contour lines and the other solid lines. Velocity is set such that each force will progress to the opposite corner of the battlefield at a constant rate.

Again the maintenance of the minimum soldier density once the two formations have passed through each other can be seen. This behaviour has not been demonstrated in any previous continuous combat model. In Figures 5.6 and 5.7, only the velocity of each force is varied with firing variables are held constant. There are two factors responsible for the differences in the final size of each force. Force v possesses a higher aimed fire constant and thus will incur a greater loss than Force u. Also, the increased velocity of both forces in Figure 5.7 minimises time in contact between them and thus firing opportunities and the subsequent loss of density. Figures 5.8a and 5.8b demonstrate that density loss occurs only when members of each force are co-located, and losses are proportional to the time in contact. The initial density profiles are:

$$u(x, y, 0) = \begin{cases} 10 & \sqrt{(x - 0.15)^2 + (y - 0.15)^2} < 0.071 \\ 0 & \text{otherwise} \end{cases}$$

$$v(x, y, 0) = \begin{cases} 10 & \sqrt{(x - 0.35)^2 + (y - 0.35)^2} < 0.071 \\ 0 & \text{otherwise} \end{cases}$$

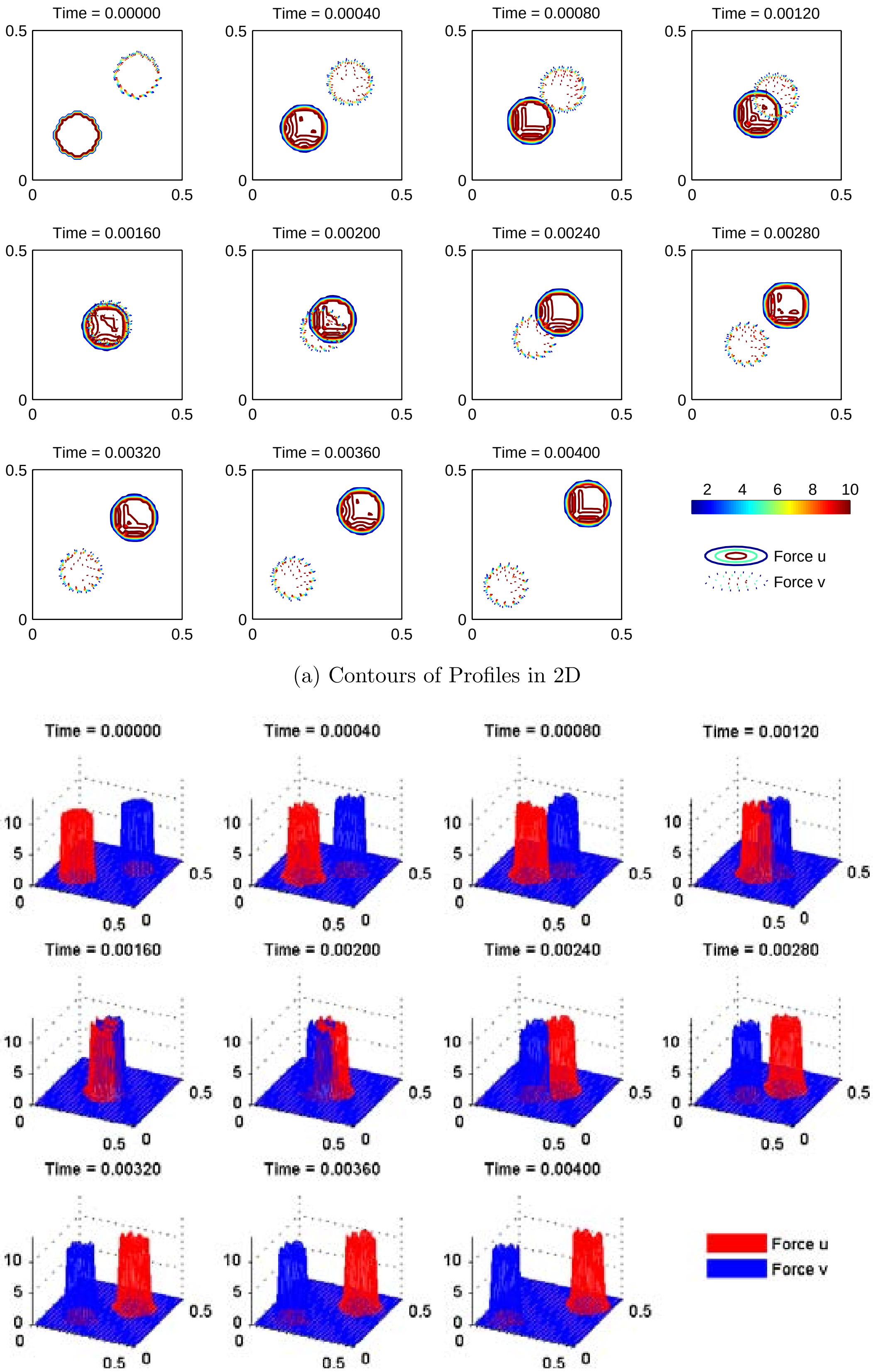
Parameters were set at $r_{u,v_{a,r}} = 5$, $D_{u,v} = 10$, $\boldsymbol{C}_u = (20, 20)$, $\boldsymbol{C}_v = (-20, -20)$, $A_{a_{u,v}} = 20$, $A_{r_{u,v}} = 2$, $d_u = 10^{-5}$, $d_v = 10^{-4}$, $\tau(t = 0) = 10^{-7}$, end time $t = 10^{-2}$, $atol = rtol = 10^{-3}$, $\Delta x = 0.01$.



(a) Contours of Profiles in 2D

(b) Graphs of profiles 3D

Figure 5.6: 2D Aimed Fire only



(a) Contours of Profiles in 2D

(b) Graphs of profiles 3D

Figure 5.7: All parameters are the same as for Figure 5.6 with the exception of $\boldsymbol{C}_u = (60, 60), \boldsymbol{C}_v = (-60, -60)$, end time $t = 4 \times 10^{-3}$.

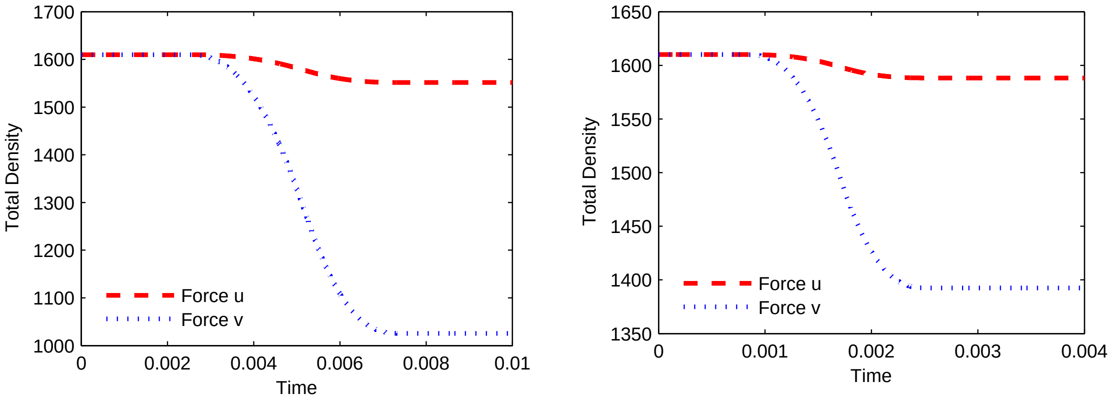


Figure 5.8: Losses for Figure 5.6 and 5.7

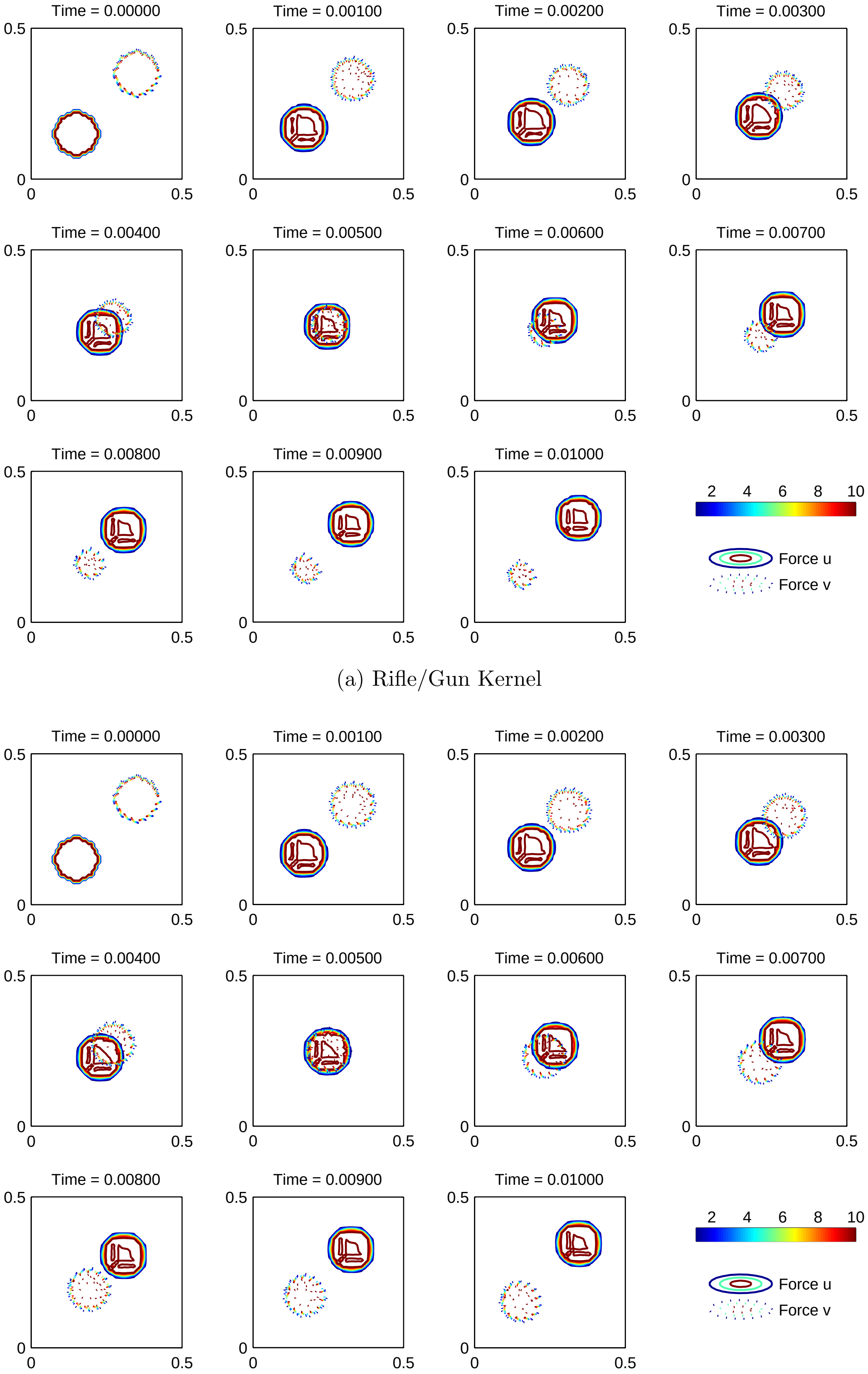
5.3.2 Area Fire Only

We now use two forms of the non-local area fire term only for $\mathbf{f_{react}}$ as detailed in Section 2.2.2, the Rifle and Artillery kernels. As expected for both kernels, each force incurs losses throughout the simulation, with the blue force sustaining the most attrition due to the difference in firing constants, $\nu_u = 8 \times 10^{-5}, \beta_u = 8 \times 10^{-8}, \nu_v = 8 \times 10^{-6}, \beta_v = 10^{-9}$. The rate of density loss seen in Figure 5.10 is of the same form as seen in Figure 4.8 due to the form of the Area Fire terms (4.11) and (4.12). The initial density profile is:

$$u(x, y, 0) = \begin{cases} 10 & \sqrt{(x - 0.15)^2 + (y - 0.15)^2} > 0.071 \\ 0 & \text{otherwise} \end{cases}$$

$$v(x, y, 0) = \begin{cases} 10 & \sqrt{(x - 0.35)^2 + (y - 0.35)^2} > 0.071 \\ 0 & \text{otherwise} \end{cases}$$

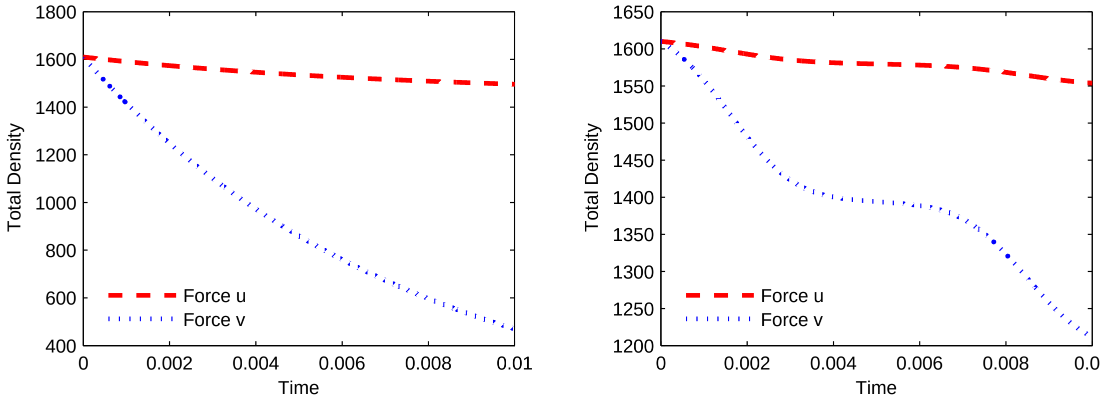
Parameters were set at $r_{u,v_{a,r}} = 5$, $D_{u,v} = 10$, $\boldsymbol{C}_u = (20, 20)$, $\boldsymbol{C}_v = (-20, -20)$, $A_{a_{u,v}} = 20$, $A_{r_{u,v}} = 2$, $\beta_u = 8 \times 10^{-8}$, $\beta_v = 8 \times 10^{-9}$, $\nu_u = 8 \times 10^{-5}$, $\nu_v = 8 \times 10^{-6}$, $\tau(t = 0) = 10^{-7}$, end time $t = 10^{-2}$, $atol = rtol = 10^{-3}$, $\Delta x = 0.01$. For Figure 5.9b the optimal range r_{op} is set at 20.



(a) Rifle/Gun Kernel

(b) Artillery Kernel

Figure 5.9: 2D Area Fire only showing different kernels



(a) Rifle/Gun Kernel Losses

(b) Artillery Kernel Losses

Figure 5.10: Losses for Figure 5.9a and Figure 5.9b respectively.

As expected, there is a continuous loss of density seen throughout the simulation using the Rifle/Gun kernel and in comparison, two distinct periods of density loss using the Artillery kernel. This corresponds to those approach and retreat periods of time when the two forces were within the optimal range. This effect can be difficult to see in the accompanying movies however Figure 5.10 demonstrates these effects quite clearly.

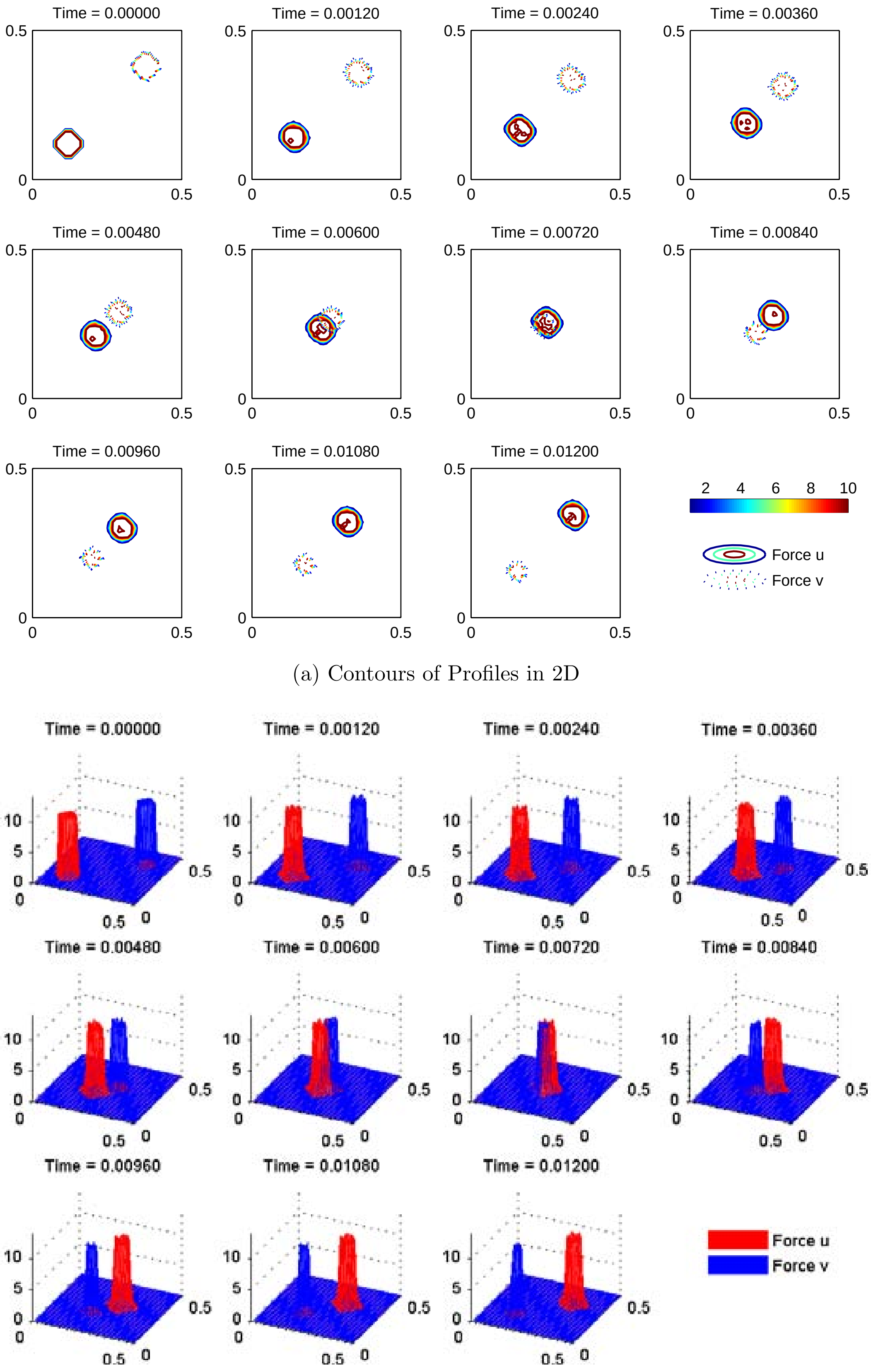
5.3.3 Aimed and Area Fire

We now investigate the full Equations (2.6) and (2.7) and begin by using the Rifle kernel for the Area Fire term. Results found after the introduction of the area fire term are as expected. Losses are gradually incurred due to the non-local area fire term and increase rapidly once the two forces meet and the aimed fire term becomes significant as shown in Figure 5.12. Maintenance of the desired internal density is also observed in Figure 5.11. The initial density profiles are:

$$u(x,y,0) = \begin{cases} 10 & \sqrt{(x-0.12)^2+(y-0.12)^2} < 0.071 \\ 0 & \text{otherwise} \end{cases}$$

$$v(x,y,0) = \begin{cases} 10 & \sqrt{(x-0.38)^2+(y-0.38)^2} < 0.071 \\ 0 & \text{otherwise} \end{cases}$$

Parameters were set at $r_{u,v_{a,r}} = 5$, $D_{u,v} = 5$, $\boldsymbol{C}_u = (20, 20)$, $\boldsymbol{C}_v = (-20, -20)$, $A_{a_{u,v}} = 5$, $A_{r_{u,v}} = 0.5$, $d_u = 2 \times 10^{-6}$, $d_v = 2 \times 10^{-5}$, $\beta_u = 8 \times 10^{-9}$, $\beta_v = 8 \times 10^{-8}$, $\nu_u = 0.3$, $\nu_v = 0.4$, $\tau(t = 0) = 10^{-7}$, end time $t = 1.2 \times 10^{-2}$, $atol = rtol = 10^{-3}$, $\Delta x = 0.01$.



(a) Contours of Profiles in 2D

(b) Graphs of profiles 3D

Figure 5.11: 2D Aimed and Area Fire, Rifle Kernel.

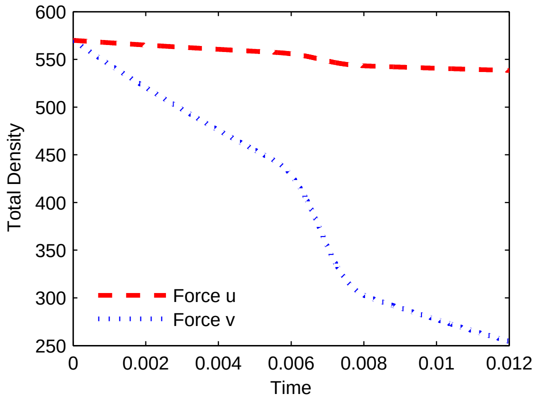
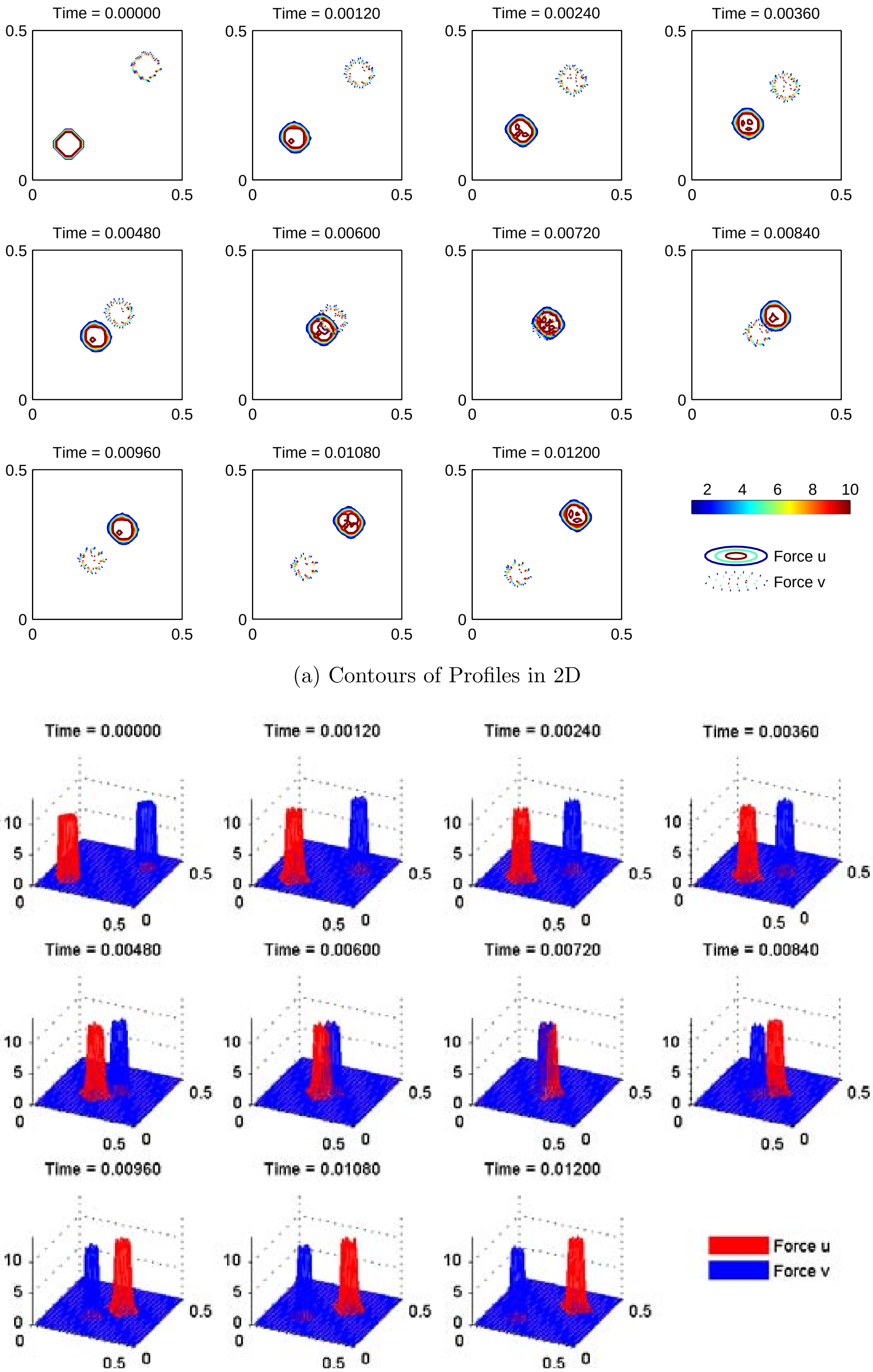


Figure 5.12: Losses for Figure 5.11

Now using the Artillery kernel as the Area Fire term and using the same parameters as for Figure 5.11 with the addition of the optimal range set to 20.



(a) Contours of Profiles in 2D

(b) Graphs of profiles 3D

Figure 5.13: 2D Aimed and Area Fire, Artillery Kernel

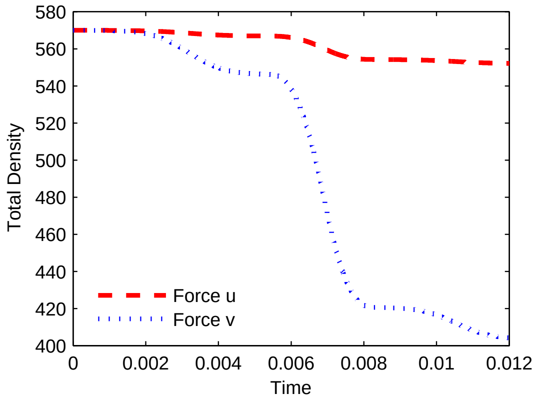


Figure 5.14: Losses for Figure 5.13.

There are three distinct periods of density loss as seen in Figure 5.14. These correspond to the two aimed firing effects as the forces come into range during the advancing period and the retreating period, and the third is due to aimed firing during overlap of the force profiles. Effectively we can see a superposition of the form of aimed fire losses in Figure 5.7 and artillery kernel area fire losses of Figure 5.10b.

We have now developed a set of partial differential equations showing the stable aggregation of forces with sharply defined edges that interact through either simple, non-local or a combination of both attrition terms. These produce similar attrition results to previously published research yet model cohesive movement throughout the simulation.

5.4 Comparison to Previous Research

Similarly for Section 4.2.3.1 we compare our results with the simple case of Frontal Attack as published in [44]. Note the distinct smearing of the profile due to diffusion and the overall retention of the initial bivariate Gaussian form of the force distributions chosen in order to simplify numerical analysis. Only the aimed fire terms were included in these results.

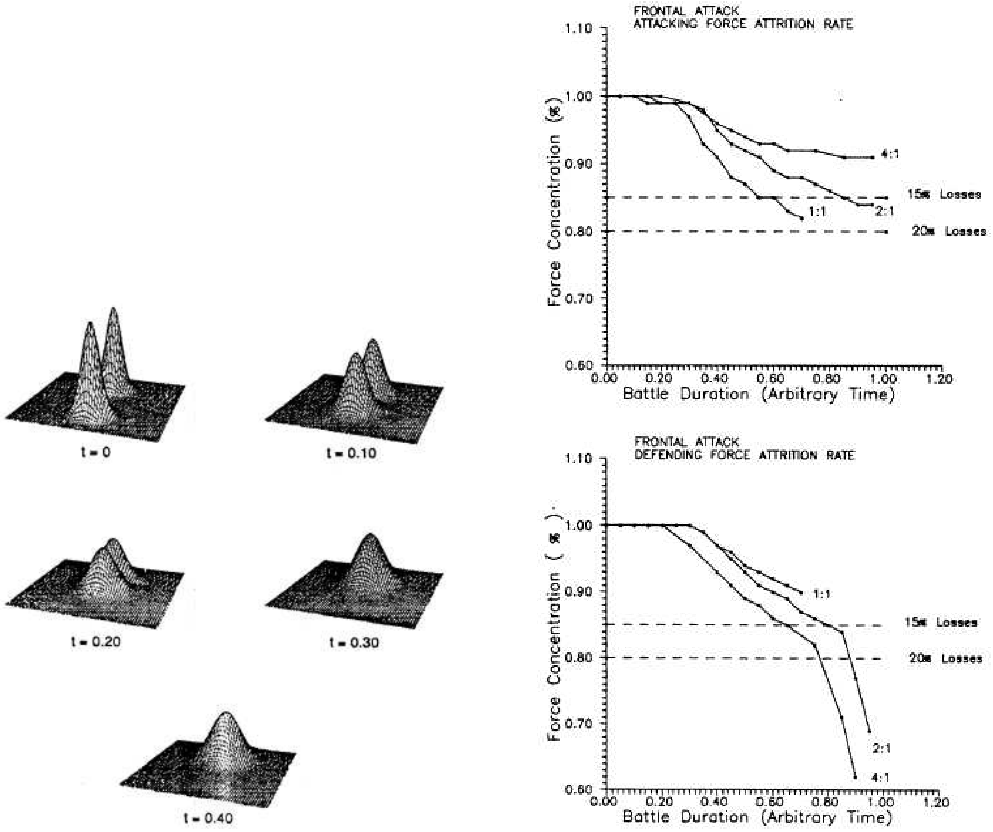


Figure 5.15: Frontal Attack Results reproduced from [44].

For the losses incurred, we concentrate on the force ratio results for 1 : 1 in Figure 5.15b. As the forces remain in contact and thus incur losses from approximately $t = 0.2$ onward, we note the casualty rate is of the same form as seen in Figures 5.8a and 5.8b for the period over which the forces were in contact.

Again, as with the one dimensional case, the force distributions show no signs of achieving constant interior densities nor well defined edges as we expect due to the simplistic representation of the spatial aspects of force movement.

5.5 Obstacles in the domain

In order to mimic the larger range of scenarios available in ISAAC and other cellular automata based wargames, obstacles were added to the model. These obstacles may represent physical boundaries such as walls, mountain ranges, lakes, rivers, unsuitable terrain, or any type of obstruction to movement. Areas where electronic or communications equipment become inoperable may also be represented this way.

We implement boundaries through the calculation of the flux function so that the flux is set to zero at the object boundaries. Using one dimension as an example, for $v(x_i, t) \geq 0$ in cell i where the cell $i+1$ forms part of the boundary in Equation 3.10:

$$F_{i+\frac{1}{2}} = 0 \tag{5.2}$$

and similarly for $v(x_i, t) < 0$ in cell i where the cell $i-1$ forms part of the boundary,

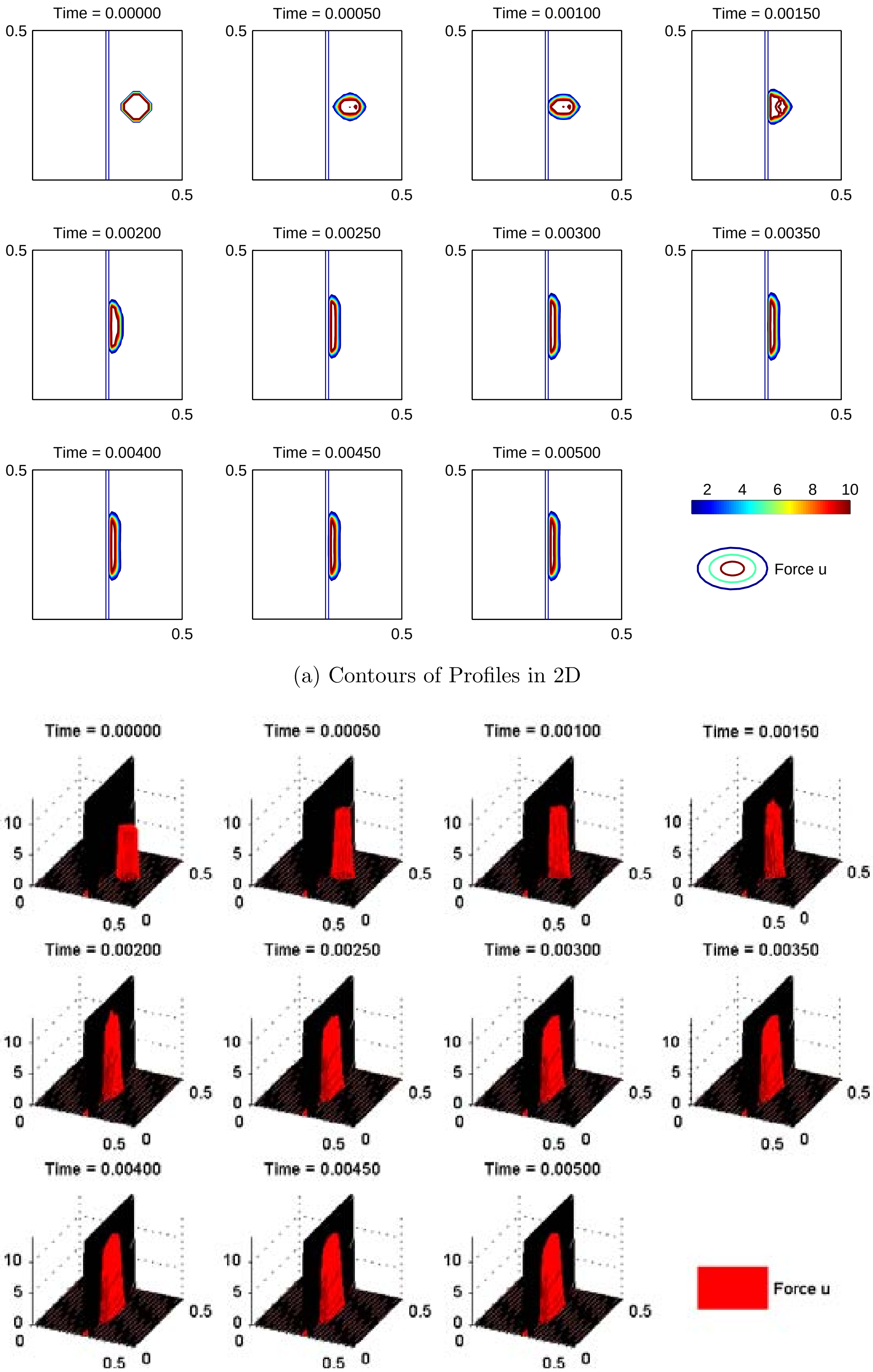
$$F_{i-\frac{1}{2}} = 0. \tag{5.3}$$

5.5.1 Wall Obstacle

Firstly, an impassable wall shaped obstacle is tested. The obstacle in this scenario represents a vertical wall dissecting the length of the domain with width 2 at $0.25 \leq x \geq 0.26$. The initial force density profile is:

$$u(x, y, 0) = \begin{cases} 8 & \sqrt{(x-0.15)^2 + (y-0.15)^2} < 0.071 \\ 0 & \text{otherwise} \end{cases}$$

Parameters were set at $r_{a,r} = 5, D = 5, \boldsymbol{C} = (60, 60), A_a = 5, A_r = 0.5, \tau(t = 0) = 10^{-9}$, end time $t = 5 \times 10^{-3}$, $atol = rtol = 10^{-3}, \Delta x = 0.01$.



(a) Contours of Profiles in 2D

(b) Graphs of profiles 3D

Figure 5.16: Wall Obstacle.

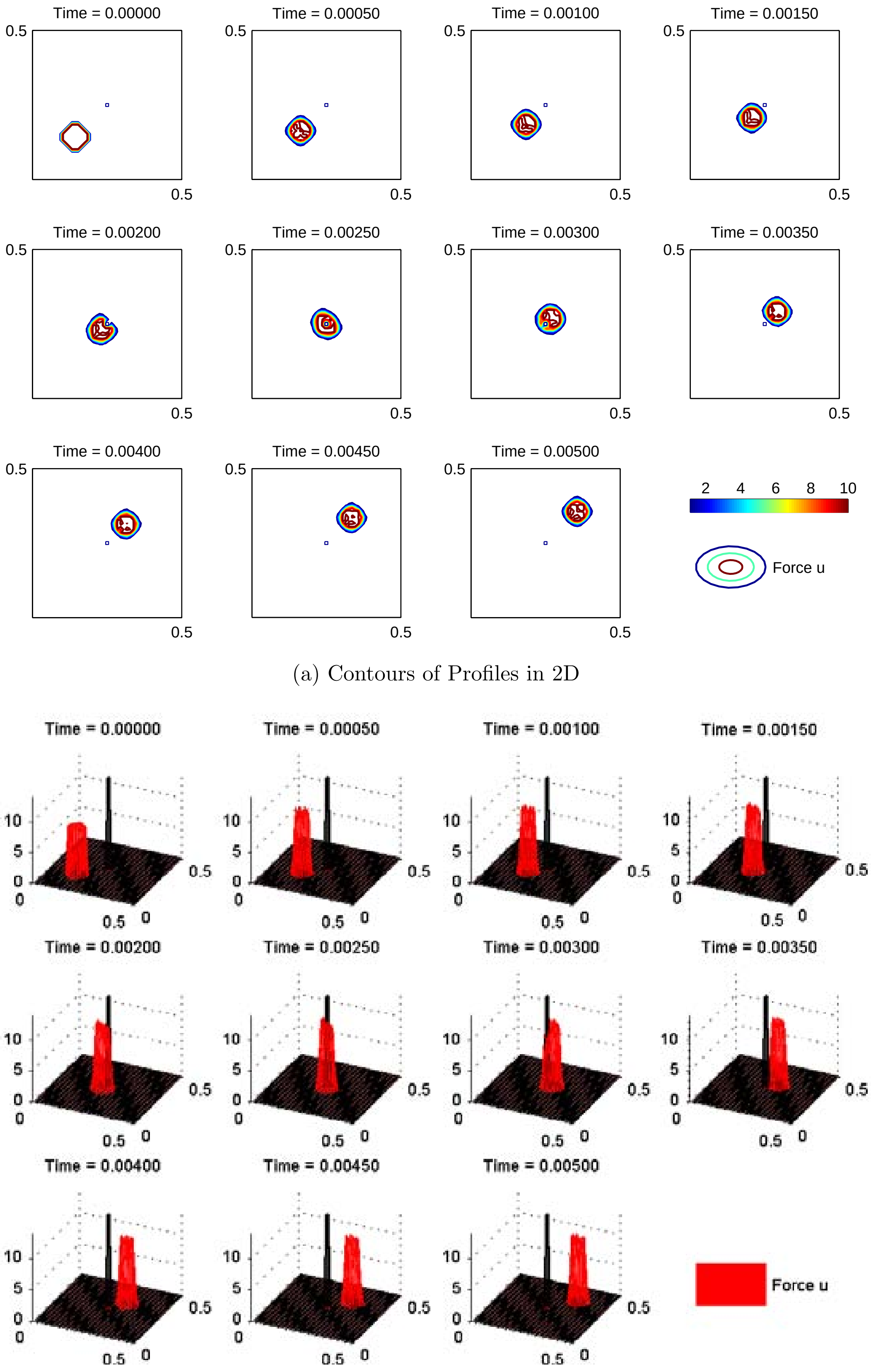
Note that as the force impacts against the wall, an increase in density due to following forces trajectory continuing directly on towards the wall. This increases the density at the wall to a higher level than A_a/A_r resulting in the repulsion term becoming significant. As no flux is permitted through the wall, this repulsion results in the force flowing along the wall in both directions until the force forms a thin strip.

5.5.2 Small Square Obstacle

Next, a small square obstacle was placed in the middle of the troop's path located at $[0.25, 0.26] \times [0.25, 0.26] \subset \Omega$. The initial density profile is:

$$u(x, y, 0) = \begin{cases} 8 & \sqrt{(x - 0.15)^2 + (y - 0.15)^2} < 0.071 \\ 0 & \text{otherwise} \end{cases}$$

Parameters were set at $r_{a,r} = 5, D = 5, \boldsymbol{C} = (60, 60), A_a = 5, A_r = 0.5, \tau(t = 0) = 10^{-9}$, end time $t = 5 \times 10^{-3}$, $atol = rtol = 10^{-3}, \Delta x = 0.01$.



(a) Contours of Profiles in 2D

(b) Graphs of profiles 3D

Figure 5.17: Small Square Obstacle.

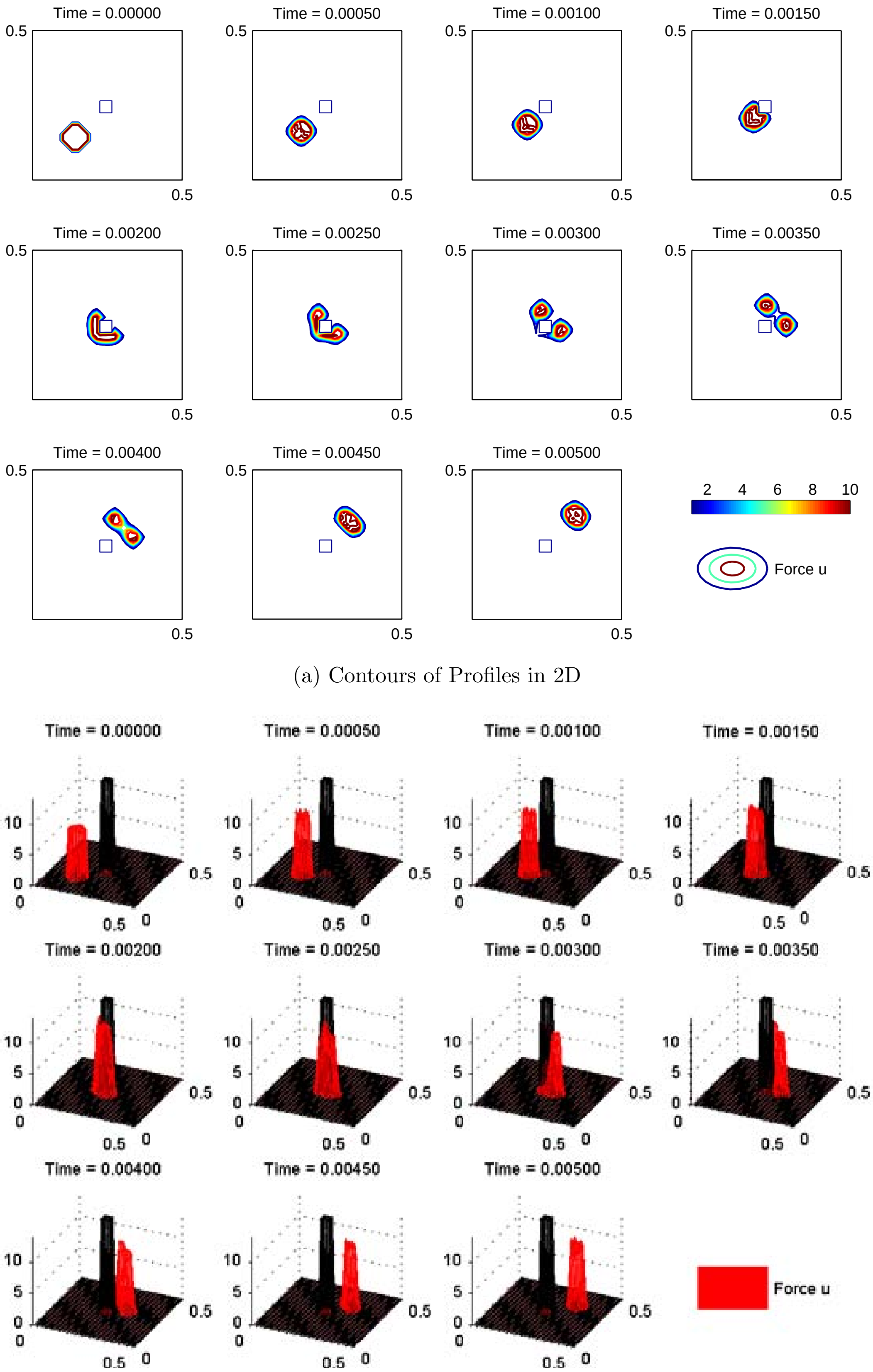
In comparison to the force profile, this obstacle is quite small and the effects to force traversal are minimal. The range of attraction is much greater than the separation created between both arms of the force profile that are formed as it moves around the obstacle. Due to this effectiveness of the attraction force, one main profile is reformed on the opposite side of the obstacle. We now gradually increase the size of the square.

5.5.3 Medium Square Obstacle

The obstacle is expanded to a larger square located at $[0.23, 0.27] \times [0.23, 0.27] \subset \Omega$. Note the separation of the force into halves and subsequent reformation as the two halves remain inside the attraction/repulsion ranges. The initial density profile is:

$$u(x, y, 0) = \begin{cases} 8 & \sqrt{(x - 0.15)^2 + (y - 0.15)^2} < 0.071 \\ 0 & \text{otherwise} \end{cases}$$

Parameters were set at $r_{a,r} = 5, D = 5, \boldsymbol{C} = (60, 60), A_a = 5, A_r = 0.5, \tau(t = 0) = 10^{-9}$, end time $t = 5 \times 10^{-3}$, $atol = rtol = 10^{-3}, \Delta x = 0.01$.



(a) Contours of Profiles in 2D

(b) Graphs of profiles 3D

Figure 5.18: Medium Square Obstacle.

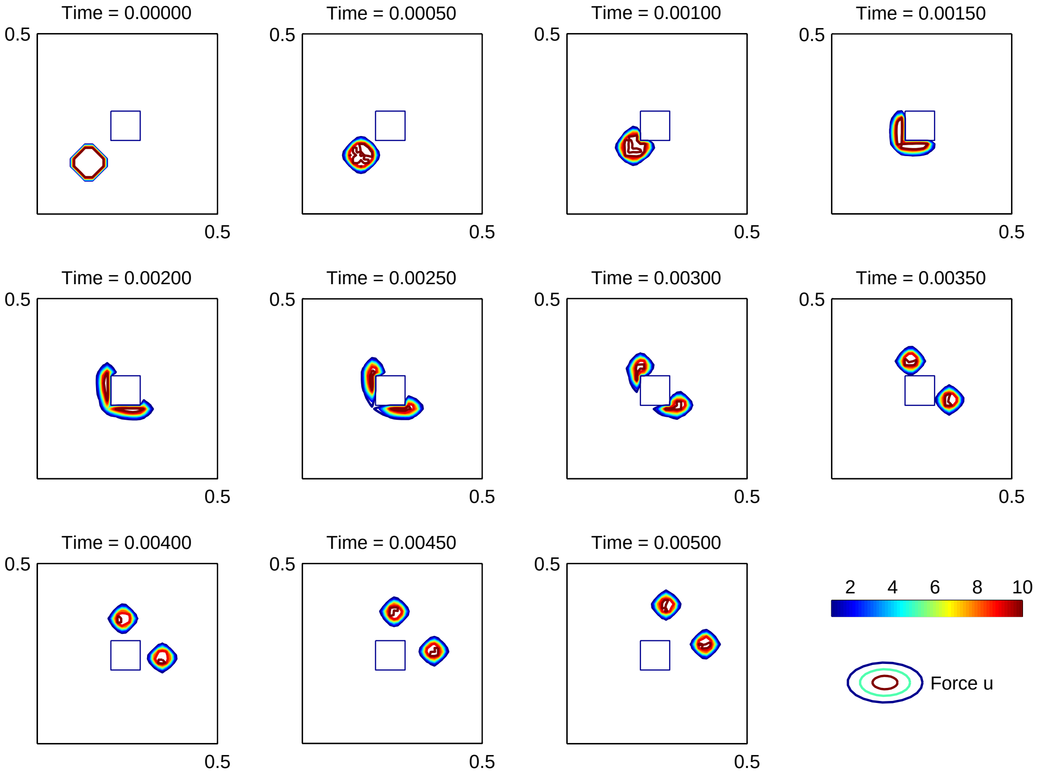
The effect of attraction distance on the force reforming is accentuated by the effective separation of the force in two. A sufficient range of attraction again results in the reformation of the force.

5.5.4 Large Square Obstacle

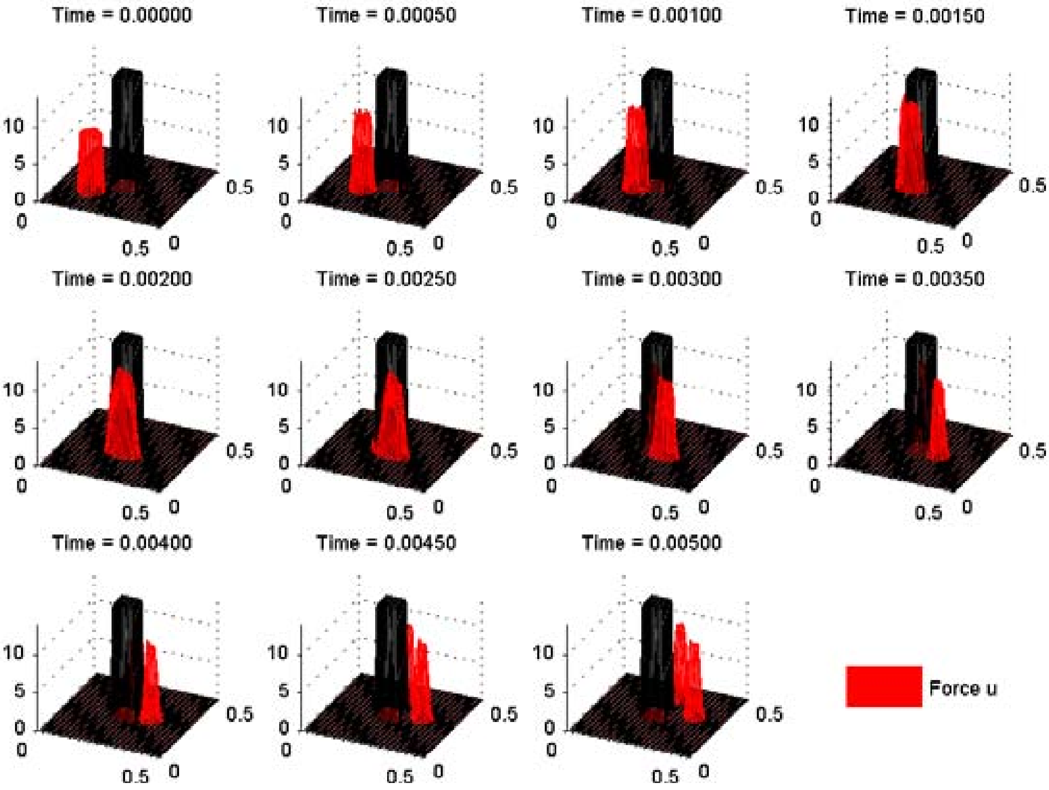
Extending this concept further, the square obstacle is increased such that each half of the force lies outside sensor range after passing around the obstacle located at $[0.22, 0.28] \times [0.22, 0.28] \subset \Omega$.The initial density profile is:

$$u(x, y, 0) = \begin{cases} 8 & \sqrt{(x-0.15)^2 + (y-0.15)^2} < 0.071 \\ 0 & \text{otherwise} \end{cases}$$

Parameters were set at $r_{a,r} = 5, D = 5, \boldsymbol{C} = (60, 60), A_a = 5, A_r = 0.5, \tau(t = 0) = 10^{-9}$, end time $t = 5 \times 10^{-3}$, $atol = rtol = 10^{-3}, \Delta x = 0.01$.



(a) Contours of Profiles in 2D



(b) Graphs of profiles 3D

Figure 5.19: Large Square Obstacle.

As expected, the two halves do not reform after passing the obstacle and form independent troop profiles.

The capability of a force to navigate around an obstacle highlights one of the differences between this model and those of Protopopescu and Spradlin. Obstacle avoidance was approximated by Protopopescu in [44] in a predetermined way. Time dependency of the convection term was artificially constructed so that the manoeuvring force effectively transits along a predefined route. In this way, various tactics such as the turning movement, envelopment and infiltration were achieved. Spradlin omitted the inclusion of obstacles altogether. Here we have shown a different method of facilitating a form of autonomous navigation throughout the domain. Although flow around obstacles is well researched and represented in fluid dynamics, this is the first time this has been included in a continuous combat model.

Now that a functional two dimensional partial differential equations based combat model as been established, we continue with comparisons to the cellular automata wargame ISAAC.

Part II

Comparison with the Cellular Automaton Model ISAAC

In Chapter 6 we describe the cellular automaton model ISAAC and a selection of scenarios its author has published to demonstrate its range of capabilities and note its ability to produce distinctive behaviours. We then show how our simulation with slight modifications can reproduce similar behaviours. Thus we show that a PDE model can perform the combat modelling tasks of a state-of-the-art CA model while being easier to understand and analyse. This can have great advantages when seemingly unexplainable or novel behaviour is seen in CA modelling results that may otherwise be attributed to a form of intelligence.

In Chapter 7 we investigate the dynamics of our model through a series of simple scenarios and attempt to reproduce these behaviours as closely as possible in ISAAC. These comparisons highlight some of the limitations in the design of ISAAC and the implications this has for the range of results expected.

Chapter 6

ISAAC

6.1 Agent-Based Wargames

There are several cellular automaton or agent-based wargaming models familiar to the military researcher. The two most common examples of which are ISAAC (Irreducible Semi-Autonomous Adaptive Combat) and MANA (Map-Aware Non-uniform Automata). Rather than deriving separate continuous equivalents to these and other military wargames through the application of coarse graining techniques, we take the published ISAAC scenarios that demonstrate the types of behaviour of interest to military researchers as representative of these CA wargame models, and compare these to our equations. Through these comparisons, we are able to determine those physically meaningful terms that generate behaviour of interest rather than concentrating solely on parameter comparisons.

6.2 ISAAC

In this chapter we describe the ISAAC simulation, giving a overview of the basic functioning of the model and how this relates to our equations. A selection of appropriate (genetic algorithms absent) published ISAAC scenarios are chosen for comparison. As some input files were missing from the ISAAC software installation package, we determine the input variables required to produce the phenomenon observed in the published results. *All* ISAAC results in the figures presented were generated by us using the necessary parameters to produce similar results to those published in [17, 18]. Then for each of these scenarios, parameters required for our model to produce similar behaviour are determined. This results in an alteration of the velocity terms of our equations to account for the effects of meta-personality. Comparisons between the discrete and continuous models are then made.

We highlight the difficulties associated with relying on atypical instances of the ISAAC scenarios to drive analysis, the effects of spatial asymmetry produced by the inherent randomness of ISAAC.

6.3 Model Description

ISAAC, Irreducible Semi-Autonomous Adaptive Combat, is a multi-agent based simulation in the style of a cellular automaton. A cellular automaton (CA) is essentially a lattice where information located on the nodes propagates from its position at each time step based on a set of defined rules. Ilachinski [18] asserts ISAAC differs from a traditional CA in that agents rather than information move throughout the lattice and that rule sets may adapt over time. In the comparison ISAAC scenarios, however, rule sets remain fixed throughout the simulation.

In ISAAC, each agent represents a simplified soldier. Soldiers are collectively grouped into “Forces” and act according to a user defined rule set for that force. That is, all agents of a Force are *homogeneous* with respect to their rule set definition. In order to create a truly *heterogeneous* battlefield of agents, each agent must be assigned to a Force consisting only of that agent. As this could result in the creation of a large numbers of Forces and is not generally practical, two opposing Forces are usually used. Additional to the rule sets, interaction in the form of weapons effects determines agent attrition.

Agents are defined to exist in three health states, alive, injured and killed and can obviously only exist in one of these states. For each time step the position for each agent is determined according to the application of the previous time step data to the rule set. Following this positional update, attrition is calculated based on the defined hit/kill probabilities. Attrition is applied to a specified maximum number of enemy agents (from zero to all agents) within a specified range.

In practice, usually two forces are defined and are typically referred to as Red and Blue with injured agents taking the same parameter values as uninjured agents.

6.3.1 Penalty Function

Movement of ISAAC agents (ISAACAs) is determined by minimisation of a penalty function calculated at each time step. Proximity of other ISAACAs, both friendly and enemy, and proximity of each force's goals are defined as the six *personality weights* used to calculate the penalty at each location within the ISAACA's movement range r_M. Note that the movement range and all other ranges used in ISAAC is a square, not a circle, of radius r_M.

Own force goals w_5 are usually located at the starting side or corner of the domain/battlespace of their respective force, with the opposition force goals w_6 located at the opposite side. As the proximity of an agent to the opposition goal increases, so the effect of the $\mathbf{w_6}$ term increases, increasing the speed at which it approaches the goal. Usually the value of $\mathbf{w_5}$ is set to zero, thus the own goal has no effect on an agent.

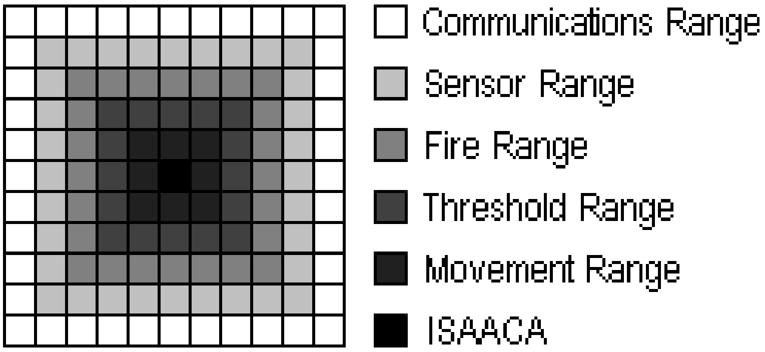


Figure 6.1: Ranges used in ISAAC

Personality weights:

Parameter	Description
$\mathbf{w_1}$	alive friendly
$\mathbf{w_2}$	alive enemy
$\mathbf{w_3}$	injured friendly
$\mathbf{w_4}$	injured enemy
$\mathbf{w_5}$	friendly flag
$\mathbf{w_6}$	enemy flag

Table 6.1: ISAAC Personality Weights.

The form of the penalty function is:

$$\begin{aligned} Z(x,y) = \quad & \frac{w_1}{\sqrt{2}r_f N_{f_{alive}}} \sum_{f_{alive};i} d[i;(x,y)] + \frac{w_2}{\sqrt{2}r_e N_{e_{alive}}} \sum_{e_{alive};i} d[i;(x,y)] + \\ & \frac{w_3}{\sqrt{2}r_f N_{f_{inj}}} \sum_{f_{inj};i} d[i;(x,y)] + \frac{w_4}{\sqrt{2}r_e N_{e_{inj}}} \sum_{e_{inj};i} d[i;(x,y)] + \\ & w_5 \frac{d_{new}[flag_f:(x,y)]}{d_{old}[flag_f:(x,y)]} + w_6 \frac{d_{new}[flag_e:(x,y)]}{d_{old}[flag_e:(x,y)]} \ . \end{aligned} \tag{6.1}$$

where $w_i, i = 1, 6$ are personality weights as mentioned. The scaling factors $\sqrt{2}r_f$ and $\sqrt{2}r_e$, number of ISAACAs N_i within sensor range and distances $d[i;(x,y)]$ of those N_i ISAACAs form a discrete convolution. d_{new} and d_{old} are the distances to the flags in question from each potential new position and from the original position respectively.

At each time step this penalty is calculated for all potential moves an agent may make, including the penalty for remaining in the original position. The position with the lowest valued penalty function is selected. When multiple new positions of equal minimum penalty occur, the new position is chosen randomly from this set.

Interestingly, Ilachinski refers to these six personality weights as constituting a *local* rule set, yet the influence of these weights, especially w_5 and w_6, can span the entire domain.

An example calculation:

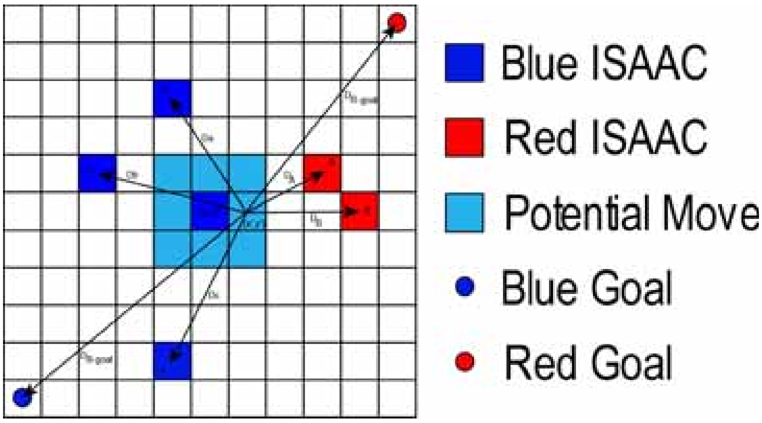


Figure 6.2: Example ISAAC scenario calculation

$$Z(x',y') = \frac{w_1}{s_{red}}(\frac{1}{3})\,[D_a + D_b + D_c] + \frac{w_2}{s_{blue}}(\frac{1}{2})\,[D_A + D_B] + w_5\frac{D_{goal_{red}}}{D^0_{goal_{red}}} + w_6\frac{D_{goal_{blue}}}{D^0_{goal_{blue}}}$$

where $D_{goal_{red}}$ and $D^0_{goal_{red}}$ are the distances from (x, y) and (x', y') to the red goal, respectively, and $D_{goal_{blue}}$ and $D^0_{goal_{blue}}$ are the distances from (x, y) and (x', y') to the blue goal, respectively.

6.3.2 Meta-Personality

In addition to this penalty function, there are six additional rules that may be implemented:

- Advance Constraint
- Cluster Constraint
- Combat Constraint
- Minimum distance to friendly ISAACAs
- Minimum distance to enemy ISAACAs
- Minimum distance to own flag.

They are collectively termed a 'meta-personality' and modify the calculation of the penalty function. These are effectively variations on some of the personality weights of (6.1) using user defined threshold and constraint ranges.

These meta-personalities will mean our PDEs as given in (2.6) and (2.7) will require modification in order to take these behaviours into account if we wish to make a direct comparison between our model and ISAAC. As these traits are not diffusion or reaction based, our $\mathbf{f_{vel}}$ term will be the section modified.

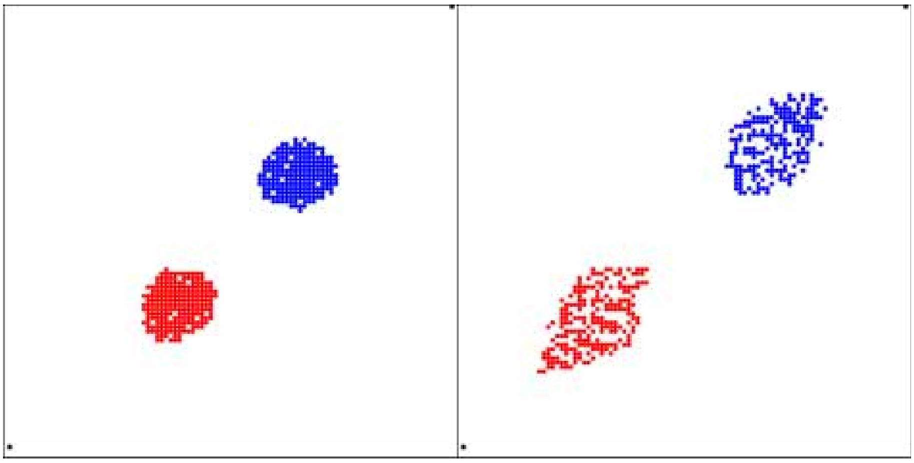
6.3.2.1 Advance Constraint

This constraint consists of specifying a threshold number of friendly ISAACAs that must be within a given ISAACA's constraint range r_C in order for that ISAACA to continue advancing toward the enemy flag. If the number of friendly forces within the range exceed this threshold, the default weight $+w_6$ is used in Equation 6.1. If this is not the case, $-w_6$ is used so that the overall desire switches to a movement away from the enemy goal. Thus sufficient troop density is required in order for advancement.

6.3.2.2 Cluster Constraint

In order for the Advance constraint to be effective, a desire to be attracted toward friendly ISAACAs to form clusters is required. Again, the threshold number of ISAACAs required to be present within the constraint range r_C is defined by the user. If this threshold is not met, the ISAACA will move in the direction of the highest density of friendly forces calculated within the sensor range. Once this threshold is reached, an ISAACA will no longer move toward friendly ISAACAs, effectively setting the parameters $w1 = w3 = 0$.

Note that this constraint combined with w_1 and w_3, mimics the attraction/repulsion kernels in our continuous equations. If the cluster constraint is not activated by the user, an *artificial* default lattice repulsion is present. Due to the density limitations of the lattice, only one agent of any type may occupy a lattice site at any time step, so that troop density cannot contract to a higher density than one troop per cell as demonstrated in Figure 6.3a.



(a) "Default" Clustering (b) Use of Cluster Constraint

Figure 6.3: Differences in clustering of forces with and without use of Cluster Constraint.

In order to maintain an average inner density of less than one troop per cell, the cluster constraint must be used in conjunction with w_1 and w_3 (Figure 6.3b).

6.3.2.3 Combat Constraint

The Combat and Advance constraints are conceptually very similar in that a minimum number of ISAACAs within a given range are required to activate a particular behaviour. This constraint is concerned with defining minimum conditions for engaging in combat with enemy ISAACAs. Two ranges (r_C) and (r_S) are used to determine the number of friendly $N_{friendly}$ and enemy N_{enemy} ISAACAs respectively. For advancement in the direction of greatest enemy concentration within the range (r_S), the threshold troop difference $\Delta_c = N_{friendly}(r_C) - N_{enemy}(r_S)$ must be exceeded. If this threshold is not exceeded, movement is determined using $w_2 = -w_{2,default}$ and $w_4 = -w_{4,default}$, where $w_{2,default}$ and $w_{4,default}$ are the default weights for moving toward alive and injured enemy ISAACAs. That is, when a numerical advantage is reached, the agents will advance along the gradient of highest enemy concentration, otherwise a numerical advantage has not been reached and the agents will retreat down the gradient of highest enemy concentration.

Setting the combat threshold to a large positive number gives a very defensive force, whereas for a large negative value a force will pursue the enemy despite the numerical disadvantage.

6.3.2.4 Other Meta-Personality and Agent Constraints

Genetic algorithms, local command and other constraints may be incorporated into ISAAC however they are not discussed in this research. Published example scenarios used as comparisons in this thesis either do not utilise these constraints or are unaffected upon their removal.

6.4 Comparisons with ISAAC Scenarios

Of those ISAAC scenarios published in [18] we investigate five which are noted by Ilachinski as demonstrating

> self-organized emergent behaviors [sic].

The scenarios utilise only the three meta-personality types as detailed above, and those agent parameters as described in the penalty function. This allows us, with slight modification to our equations, to compare our continuous model to a well accepted cellular automata based wargame.

Descriptions are based on our ISAAC run results which are similar to the published screen shots as given in [18]. Where the behaviour observed in running scenarios differs markedly from the published and assumed representative results, further descriptions are given of what is seen to be the actual representative behaviour.

Distributions of forces with initial densities ID located at c will be of the form:

$$w_i(x,y,0) = \begin{cases} \text{ID} & (\sqrt{\frac{\rho}{\rho\sqrt{2\pi}}} * e^{\frac{-|c-\mu|}{2\rho^2}}) > \text{Initial Threshold (IT)} \\ 0 & \text{otherwise} \end{cases} \quad i = u, v \quad (6.2)$$

6.4.1 Difficulties with Representative Scenarios

During the prosecution of this research, one major shortcoming of using an agent based model of this nature was highlighted - stochasticity. Those results shown in the figures contained within the scenario descriptions of [18] are not necessarily indicative of any expected behaviour. Although there is some facility for the collection of basic statistics such as average cluster size, spatial entropy or Red/Blue interpoint distance, there has been no undertaking to establish whether the given realisations are true representatives of their scenario parameters, despite being presented by Ilachinski as the expected or mean behaviour. Graham and Moyeed [11] note that Lagrangian models are akin to experiments rather than any theoretical undertaking and provide a framework for establishing reliability of such results. We do not pursue the application of such a framework here. Rather we take the ISAAC results presented at face value while noting the frequency of realisations that match the stated behaviour, and through our continuous approach provide a type of verification in a similar vein to [10]. Had the software not been available for use in order to recreate these scenarios, the noted discrepancies between the published and seemingly representative results to the variations observed and described in this thesis would not have been found. We believe the establishment of mean behaviour of agent based models such as ISAAC and MANA should rest with their developers.

6.4.2 Classic Fronts Scenario

6.4.2.1 Description

We begin with the input file included with the Einstein Test Release Version 1.0.0.4 Beta, Build Date 2000, for the "einstein_classic_fronts" scenario which Ilachinski likens to a *clash between two viscous fluids*. The two loosely grouped forces collide and align in relatively stable long thin fronts. Attrition between these fronts and the inherent randomness of the movement updates causes a discrepancy in density at either the upper or lower point of the formations. Once this occurs, the forces are able to slowly filter around each other and proceed to their respective goals.

6.4.2.2 ISAAC Parameters and Results

We now investigate the minimum number of parameters and their values required to display the original behaviour. Firstly, the parameters for alive and injured friendly and enemy ISAACAs are made equal such that there is no distinction in behaviour of an alive or injured ISAACA. Personality weights towards friendly forces (and thus clustering effects) were set to zero and found not to affect the overall “front forming” behaviour. Switching the Advance parameter off also had no effect. The final parameter values used are in Table 6.2.

Parameters used:

Parameter	Red	Blue	Parameter	Red	Blue
Squad Size	225	225	Combat	3	3
w_1	0	0	Battlefield length	100	
w_2	50	50	Battlefield width	100	
w_3	0	0	Initial Dist Centre x	15	85
w_4	50	50	Initial Dist Centre y	50	50
w_5	0	0	Size x	25	25
w_6	5	5	Size y	25	25
r_S	5	5	Flag x	1	99
r_F	3	3	Flag y	50	50
r_T	2	2	Terrain	no	
w_M	1	1	Move sampling order	random	
Prob Hit	0.002	0.002	Fratricide	no	no
Max Sim tgts	5	5	Reconstitution	no	no
Defence Measure	1	1	Terrain	no	
Cluster	NA	NA			

Table 6.2: ISAAC Parameters for Classic Fronts Scenario. All other parameters are set to zero or *no*.

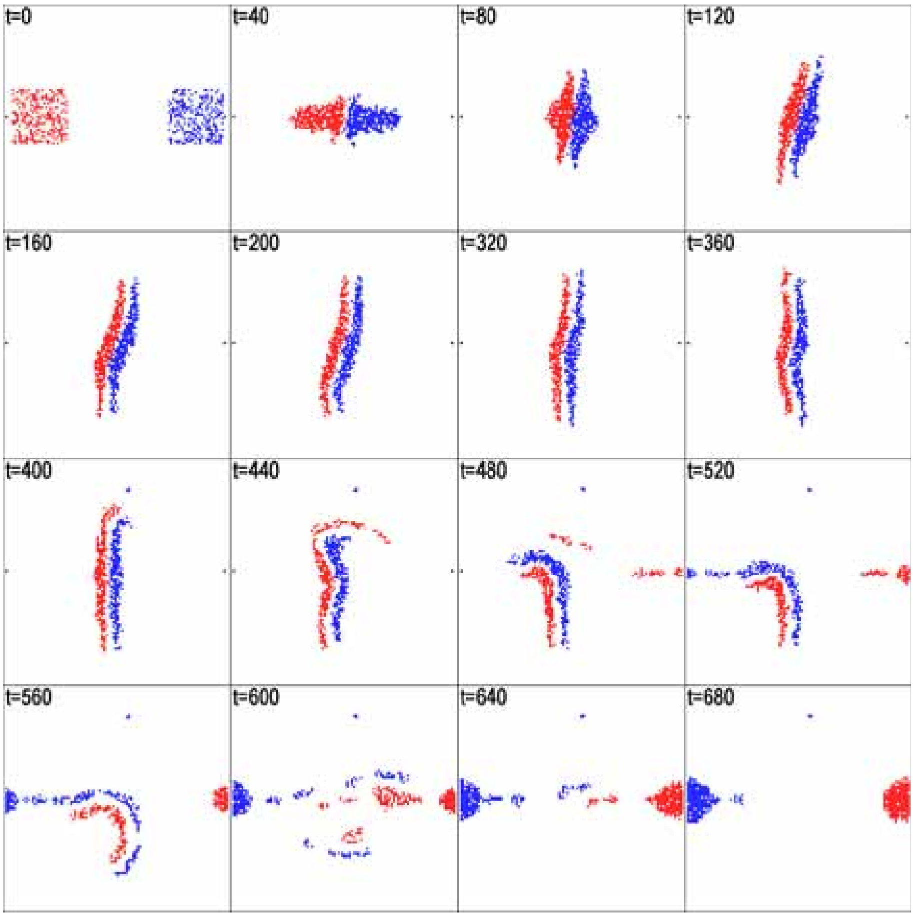


Figure 6.4: These snapshots were generated using the provided input file and demonstrates equivalent behaviour to those in [18]

6.4.2.3 PDE Modification

Our original equations are modified to reflect the addition of the combat meta-personality. This is effectively a modification of the velocity term (2.1) as we wish for only the direction of movement to be affected by the presence of the opposite force. Diffusion, inter-force attraction and repulsion are to remain unaffected.

$$\mathbf{f_{adv}} = \nabla \cdot \{u(\mathbf{C_u} u + A_a(K_a * u) - A_r u(K_r * u))\} \tag{6.3}$$

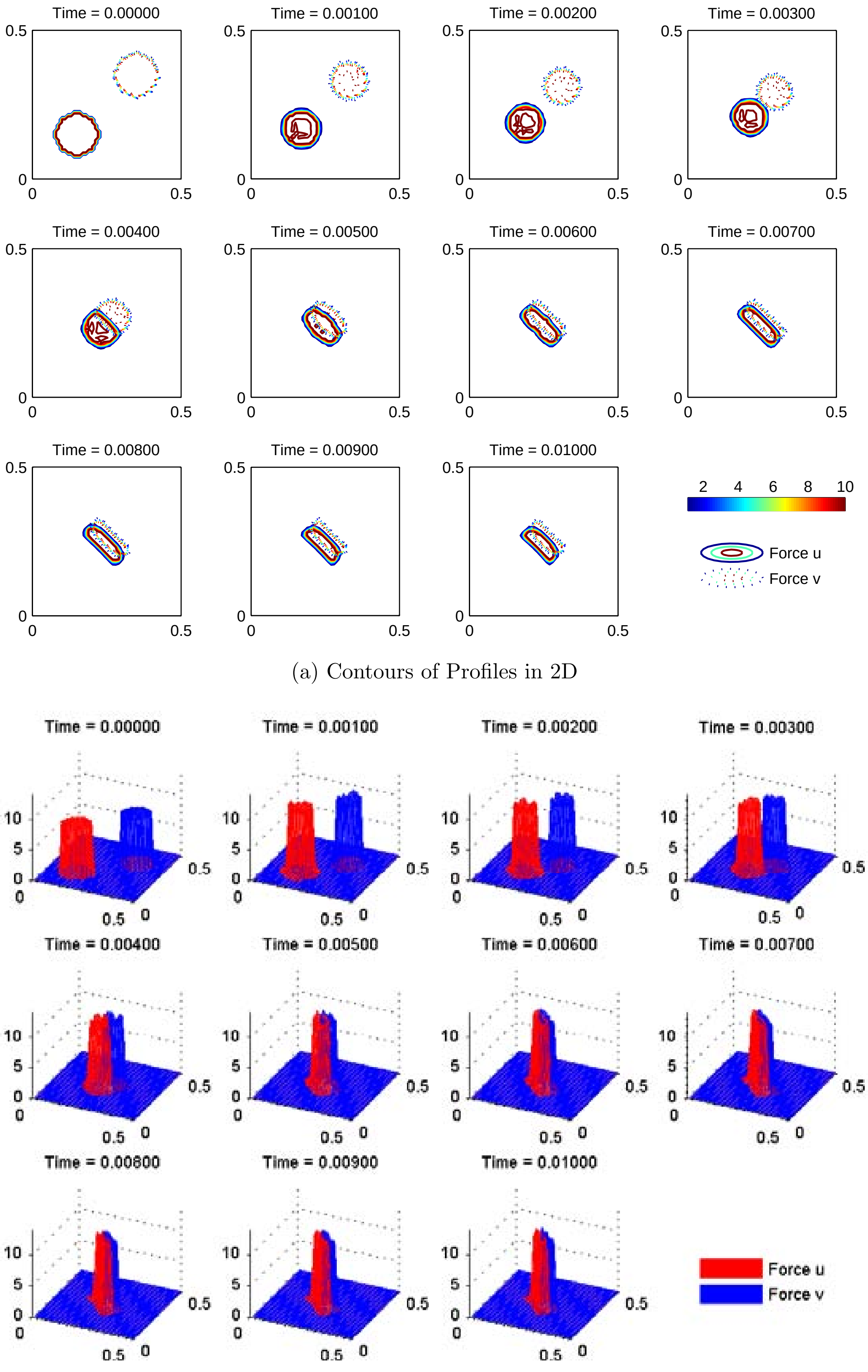
We propose the following form of the velocity term, taking force u as an example:

$$\mathbf{C} = \begin{cases} C & if\ (\int_{x-r_S}^{x+r_S}\int_{y-r_S}^{y+r_S} u\ dxdy - \int_{x-r_S}^{x+r_S}\int_{y-r_S}^{y+r_S} v\ dxdy) > \Delta c \\ \\ -C & if\ (\int_{x-r_S}^{x+r_S}\int_{y-r_S}^{y+r_S} u\ dxdy - \int_{x-r_S}^{x+r_S}\int_{y-r_S}^{y+r_S} v\ dxdy) \leq \Delta c \end{cases} \tag{6.4}$$

where $(\int_{x-r_S}^{x+r_S}\int_{y-r_S}^{y+r_S} udxdy)$ is the number of friendly forces and $(\int_{x-r_S}^{x+r_S}\int_{y-r_S}^{y+r_S} vdxdy)$ the number of enemy forces within the sensor range r_S. C is a constant and represents the overall movement toward the u or friendly flag. If a numerical advantage greater than Δc is reached, the force will proceed with a constant velocity toward its own flag. Should this numerical advantage not be attained, the velocity of the force is reversed such that it will retreat away from its flag.

6.4.2.4 PDE Results

Parameters were set at $\mathrm{ID}_{u,v} = 8$, $\rho_{u,v} = 0.05$, $\mu_u = (0.15, 0.15)$, $\mu_v = (0.35, 0.35)$, $\mathrm{IT}_{u,v} = 0.5$, $r_{a,r_{u,v}} = 5$, $D_{u,v} = 5$, $\boldsymbol{C_u} = (20, 20)$, $\boldsymbol{C_v} = (-20, -20)$, $A_{a_{u,v}} = 5$, $A_{r_{u,v}} = 0.5$, $r_{S_{u,v}} = 3$, $\Delta c_{u,v} = 100$, $attack_{u,v} = -1$, $d_{u,v} = 2 \times 10^{-6}$, $\beta_{u,v} = 8 \times 10^{-8}$, $\nu_{u,v} = 0.2$, $\tau(t = 0) = 10^{-7}$, end time $t = 10^{-2}$, $atol = rtol = 10^{-3}$, $\Delta x = 0.01$.



(a) Contours of Profiles in 2D

(b) Graphs of profiles 3D

Figure 6.5: Our Model: Classic Fronts comparison.

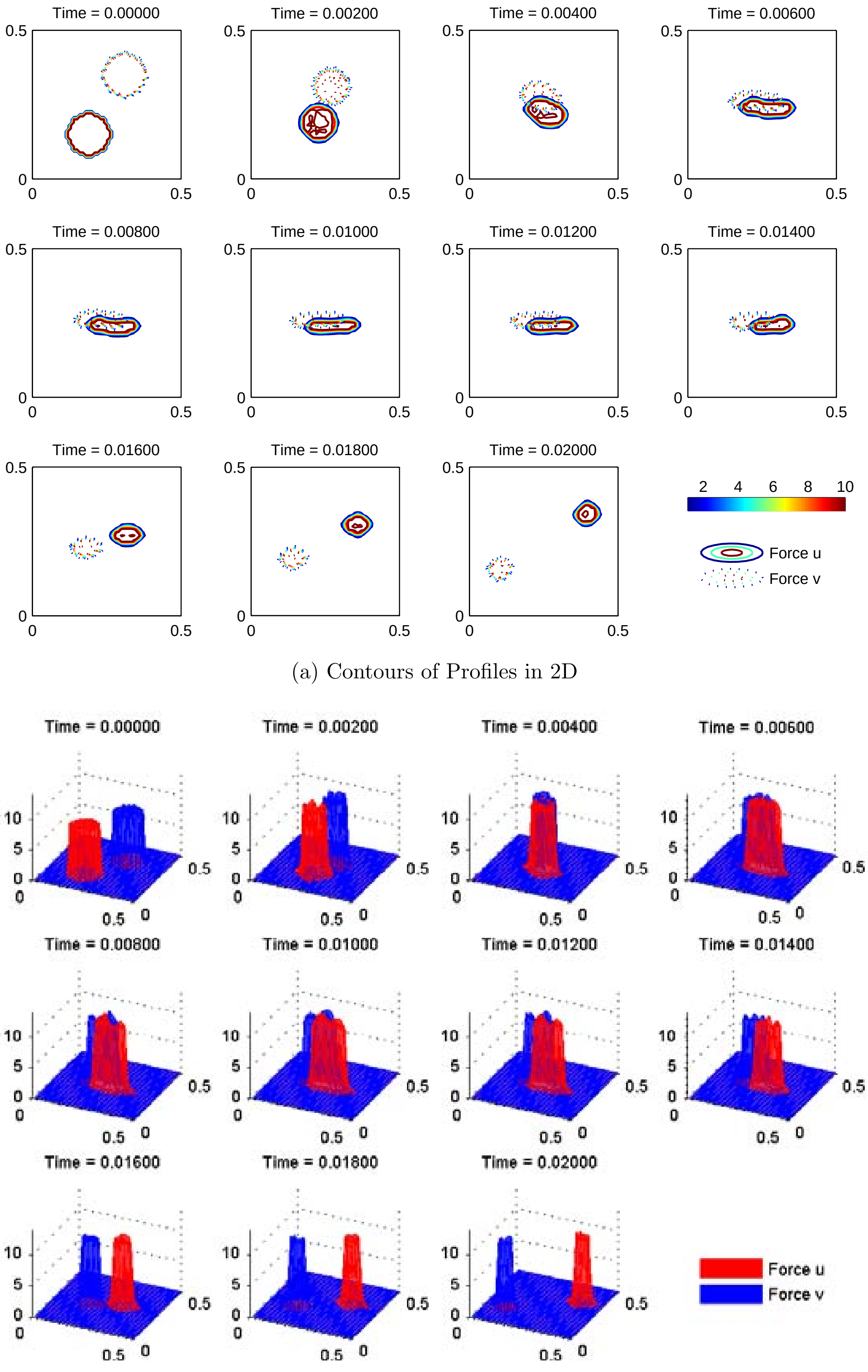
ISAAC has two areas of inbuilt randomness or stochasticity; (i) new position selection when there are multiple positions with equal penalty function values and (ii) the calculation of casualties/fatalities and updated after each positional update. This randomness results in slight differences in the spatial distribution of the two forces whereas the non-random continuous version produces equal distributions as expected. The ISAAC results in Figure 6.4 shows the forces "slipping" around each other due to these slight variations in distributions and then proceeding to their respective goals. Figure 6.5 shows that our continuous version also forms long thin fronts however these remain stationary throughout the duration of the simulation. Due to the steady reduction in both forces due to aimed fire, their densities gradually decline however the position of the fronts remain stationary. This tactic represents a classic style of attrition warfare. If the simulation is allowed to progress, both forces will eventually decline to zero. There is no randomness present to generate the slight differences or asymmetry in spatial distribution that leads to the forces manoeuvring around each other as in Figure 6.4.

Introducing spatial asymmetry through the initial density profile positions forms a type of controlled stochasticity approximation, providing a mechanism to further explore the observed ISAAC dynamics. In this case, it allows us to confirm our conjecture that asymmetry facilitates the "slippage". By effectively removing the inherent stochasticity of ISAAC, we can determine the mean behaviour of a scenario and more easily understand the effects of parameter changes.

In order to introduce an asymmetry to the continuous model so as to mimic the asymmetry seen in the ISAAC results, the initial positions of each force distribution is changed such that the velocity vectors are offset. That is, the forces will no longer collide "head on".

Parameters were set at $\mathrm{ID}_{u,v} = 8$, $\rho_{u,v} = 0.05$, $\mu_u = (0.19, 0.15)$, $\mu_v = (0.31, 0.35)$, $\mathrm{IT}_{u,v} = 0.5$, $r_{a,r_{u,v}} = 5$, $D_{u,v} = 5$, $\boldsymbol{C_u} = (20, 20)$, $\boldsymbol{C_v} = (-20, -20)$, $A_{a_{u,v}} = 5$, $A_{r_{u,v}} = 0.5$, $r_{S_{u,v}} = 3$, $\Delta c_{u,v} = 100$, $attack_{u,v} = -1$, $d_{u,v} = 2 \times 10^{-6}$, $\beta_{u,v} =$

$8 \times 10^{-8},\ \nu_{u,v} = 0.2,\ \tau(t = 0) = 10^{-7}$, end time $t = 2 \times 10^{-2}$, $atol = rtol = 10^{-3}$, $\Delta x = 0.01$.



(a) Contours of Profiles in 2D

(b) Graphs of profiles 3D

Figure 6.6: Our Model: Classic Fronts comparison with forces initially offset.

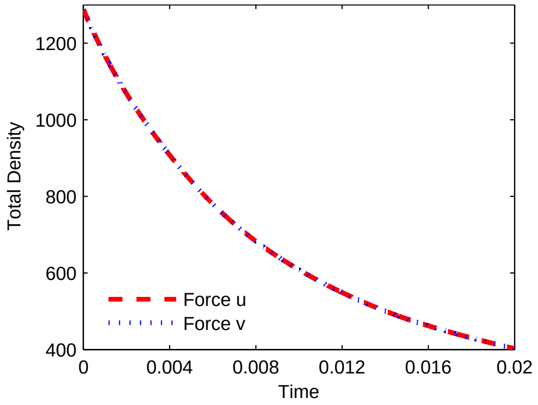


Figure 6.7: Losses for Figure 6.6

Again the formation of fronts is present however they form at a slight angle rather than vertically due to the initial offset in position. A slight oscillation in the movement of each force is visible in Figure 6.6 as they slowly rotate around each other. This is more noticeable in the movie of this scenario. Once the forces pass, circular profiles with the respective desired minimum densities are reformed. Comparing Figure 6.4 to Figure 6.6, the tight troop formation of the ISAAC agents seen prior to the forces making contact in Figure 6.4 is not seen after the confrontation. In the ISAAC results, troops stream to their respective flag in single file rather than proceeding in formation due to the effect of increasing flag proximity (w_6) on the penalty function. However, maintaining a coherent profile throughout the *entire* simulation, especially when a force is not in contact with another, is a highly desirable feature as a long thin profile may have an increased vulnerability to attack.

6.4.3 Precess Scenario

6.4.3.1 Description

This scenario is described by Ilachinski as showing a *simple example of an emergent behaviour*. Red and blue forces are quite different in personality, with red preferring to remain in close proximity while blue actively seeks the red force within sensor range. Initially the rapidly advancing blue force is loosely formed and the slower red force advances in a higher density and thus smaller formation. As the forces enter into each other's sensor range, the blue force partially surrounds the red and a slow precession begins. The red force slowly continues advancing toward its flag, constantly pursued by the blue force as shown in Figure 6.10. This precession behaviour, or rotation of both forces around an axis, as shown in the published ISAAC results and Figure 6.10 occurs in only a small percentage of the test runs. When precession was observed, both clockwise and anti-clockwise precession occurred with the forces often colliding with the domain boundaries. Offsetting the position of the blue flag slightly to the left or right did result in a higher frequency of precession observed, with precession directionality depended upon the flag offset.

Without the flag offset the usual behaviour is as follows using the initial conditions as given in Table 6.3. As the forces come into sensor range, the blue force surrounds the red until the majority of the blue force is located on the south western side of the red force. As the personality weights for engaging in combat is stronger than for continuing to the blue flag, the blue force actively pursues the tightly clustered red force as it moves toward its red flag goal. Both forces remain in coherent formations throughout the duration of the scenario as seen in Figure 6.8.

We propose that, similar to the Classic Fronts scenario, this precession behaviour arises due to the spatial asymmetry of the forces. Depending on the spatial distribution on falling into sensor range, the precession will be either clockwise or anticlockwise, and offsetting the initial positions exacerbates this asymmetry thus increasing the frequency of precession occurring.

6.4.3.2 ISAAC Parameters and Results

Unlike the Classic Fronts scenario, no input file was provided by the developer of ISAAC with the software installation, only a series of snapshots with a partially complete list of parameter values. Those unspecified values were determined through a trial and error procedure until the behaviour shown in [18] could be reproduced. In a similar manner as for the Classic Fronts scenario, we set the *Advance* and *Minimum Distance* parameters to zero and find that precession behaviour is still produced although more infrequently, suggesting that these parameters are unnecessary for this behaviour to occur.

Parameters used:

Parameter	Red	Blue	Parameter	Red	Blue
Squad Size	90	90	Cluster	10	3
w_1	25	10	Combat	4	-5
w_2	10	35	Battlefield length	100	
w_3	75	10	Battlefield width	100	
w_4	25	80	Initial Dist Centre x	10	90
w_5	0	0	Initial Dist Centre y	10	90
w_6	50	50	Size x	20	20
r_S	5	5	Size y	20	20
r_F	3	3	Flag x	1	90
r_T	2	3	Flag y	1	99
w_M	1	1	Terrain	no	
Prob Hit	0.002	0.002	Move sampling order	random	random
Max Sim tgts	All	All	Fratricide	no	no
Defence Measure	1	1	Reconstitution	no	no
Advance	0	0	Terrain	no	

Table 6.3: ISAAC Parameters for the Precess scenario. Partially supplied in the literature with the remaining values determined by the author. All other parameters are set to zero or *no*.

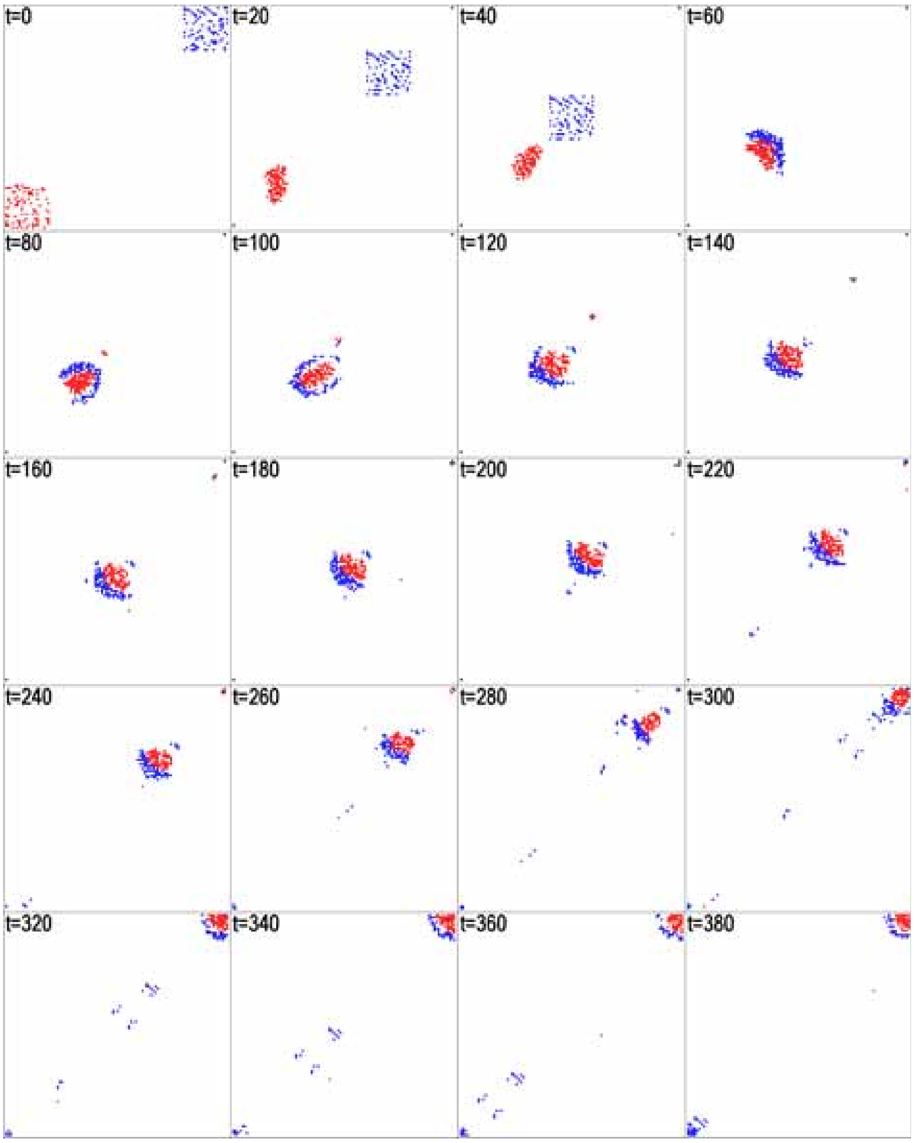


Figure 6.8: ISAAC Precess Scenario Screenshots

Figure 6.8 shows the usual results found when running the input file as detailed in Table 6.3. Note the absence of precession. As indicated above, precession in both directions can be observed with the direction determined by the spatial asymmetry generated in that particular instance.

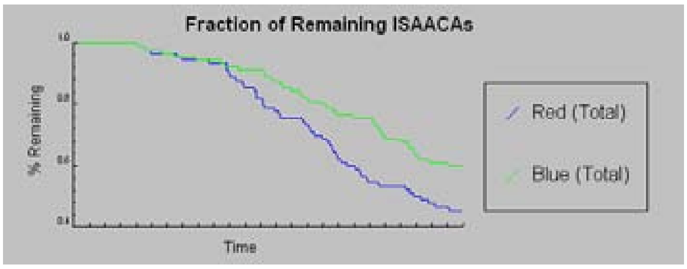


Figure 6.9: Losses for Figure 6.8.

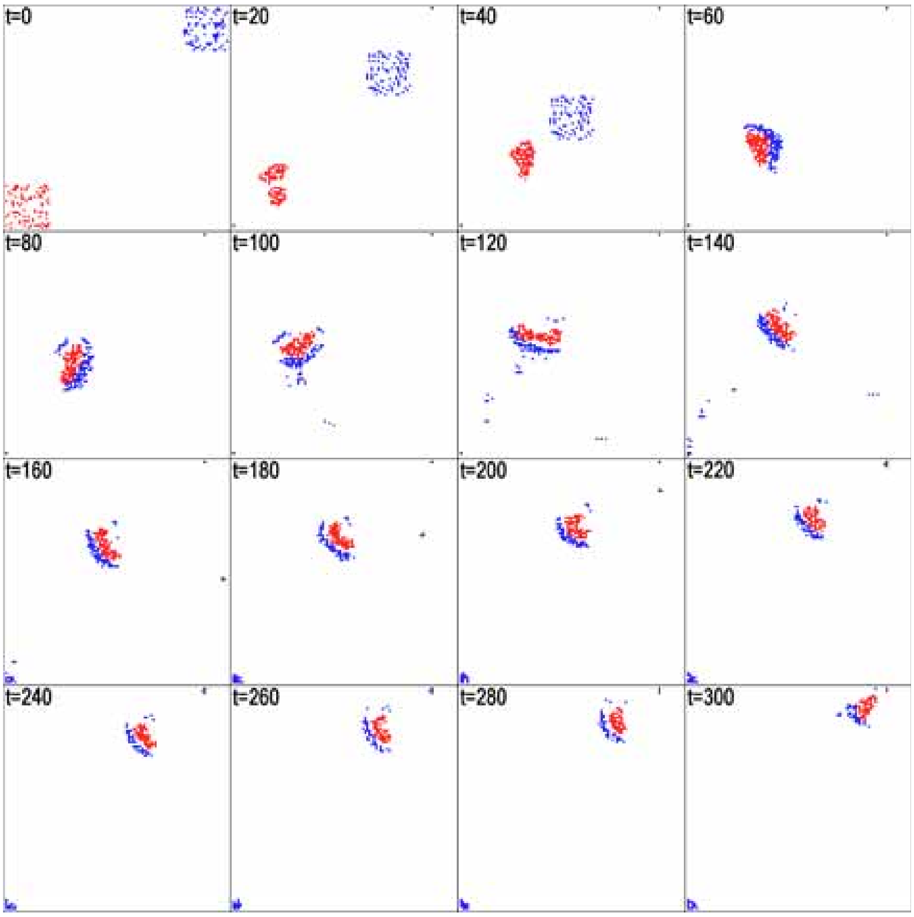


Figure 6.10: ISAAC Precess Scenario Screenshots with Artificial Offset.

By introducing an artificial offset, precession is almost always observed (Figure 6.10) and is dictated by the direction of the given offset.

6.4.3.3 PDE Modifications

We now seek to modify the velocity term in a similar way as with the Classic Fronts scenario. Our modification to the velocity term described in (6.4) does not adequately take into account the differences in offensive/defensive personalities as described in the ISAAC literature and must be expanded in order to include this. Firstly, we define the number of friendly (N_u) and enemy forces (N_v) within the *Sensor* range used to determine the *Combat* constraint by:

$$N_v = \int_{x-r_S}^{x+r_S} \int_{y-r_S}^{y+r_S} v \; dxdy; \qquad N_u = \int_{x-r_S}^{x+r_S} \int_{y-r_S}^{y+r_S} u \; dxdy; \tag{6.5}$$

In order to allow for the two different types of attacking personalities, a switch *attack* in the form of an integer of value 1 or -1 is included. This allows for a distinction between two force types - an aggressive and a defensive force. An aggressive force will move toward the enemy regardless of superiority in numbers, while a defensive force will advance only with superior numbers and will otherwise retreat. We now alter the velocity term again and arrive at the following form:

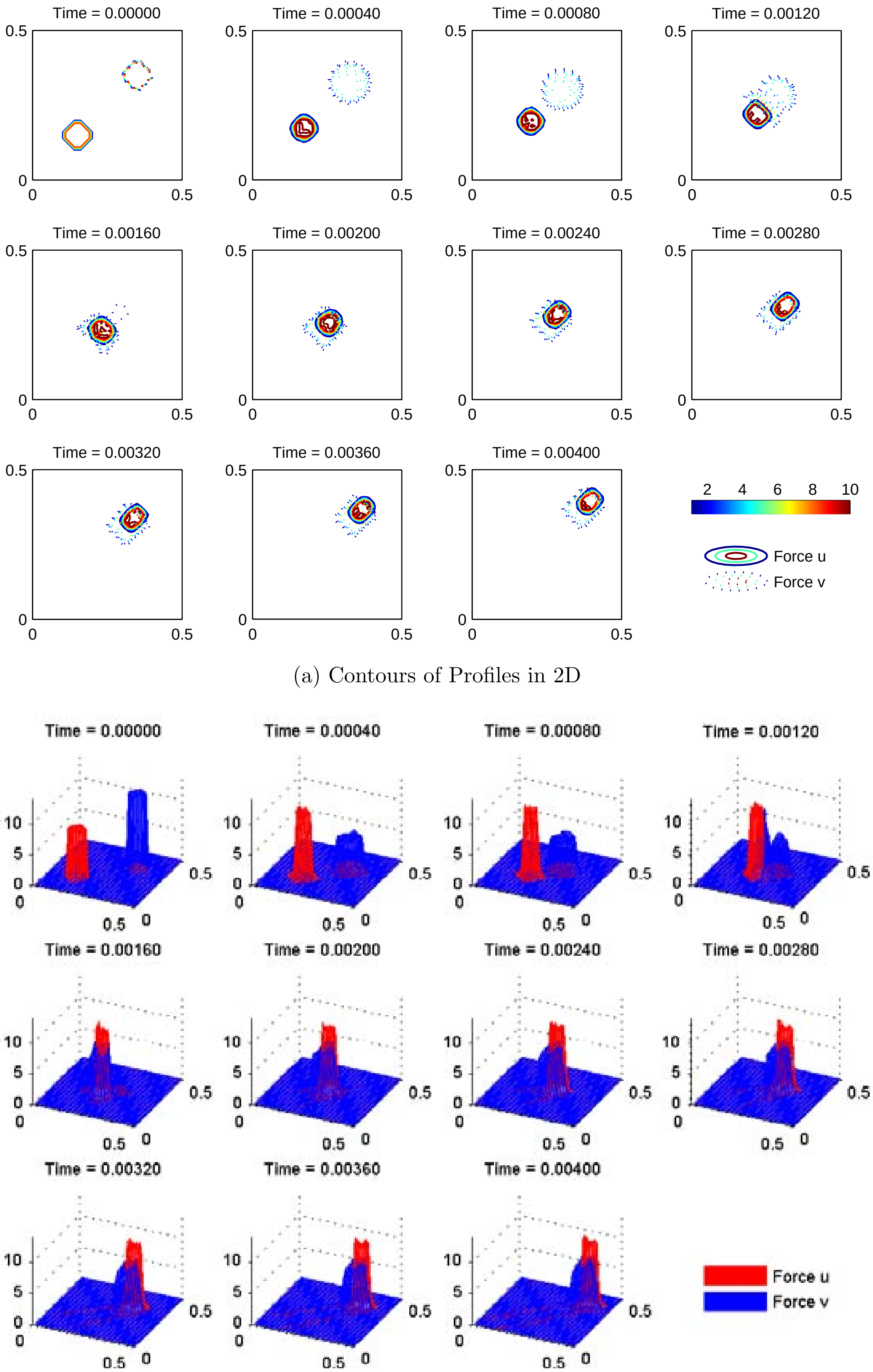
$$\mathbf{C} = \begin{cases} C + \int_{x-r_S}^{x+r_S} \int_{y-r_S}^{y+r_S} v \; dxdy & N_u - N_v \geq \Delta c \\ \\ C + attack \times \int_{x-r_S}^{x+r_S} \int_{y-r_S}^{y+r_S} v \; dxdy & N_u - N_v < \Delta c \end{cases} \tag{6.6}$$

With a numerical superiority above the given Δc, the force has an additional attraction up the gradient of greatest concentration of the opposing force. If this superiority level is not reached, this attraction is reversed, becoming a repulsion down the gradient of greatest concentration. This effectively mimics the Combat constraint of ISAAC as described in Section 6.3.2.3.

6.4.3.4 PDE Results

Figure 6.11 shows the first comparison at the ISAAC precession results and is markedly similar to Figure 6.8. Upon falling into the *Sensor* range of the smaller footprint higher density force (Force 1, coloured red), the larger footprint lower density force (Force 2, blue) rapidly moves to surround it as $attack = 1$ for Force 2. As with the ISAAC scenario, this aggressive force pursues Force 1 for the remainder of the simulation with the majority of the force located on the south western side of Force 1 and a small portion of Force 2 located on the north eastern side of Force 1.

Parameters were set at $\mathrm{ID}_u = 8$, $\mathrm{ID}_v = 12$, $\rho_{u,v} = 0.05$, $\mu_u = (0.15, 0.15)$, $\mu_v = (0.35, 0.35)$, $\mathrm{IT}_{u,v} = 1$, $r_{a,r_{u,v}} = 5$, $D_{u,v} = 5$, $\boldsymbol{C_u} = (20, 20)$, $\boldsymbol{C_v} = (-20, -20)$, $A_{a_{u,v}} = 5$, $A_{r_u} = 0.5$, $A_{r_v} = 1$, $r_{S_u} = 3$, $r_{S_v} = 7$, $\Delta c_u = 10^6$, $\Delta c_v = 4$, $attack_{u,v} = -1$, $d_{u,v} = 2 \times 10^{-6}$, $\beta_{u,v} = 8 \times 10^{-8}$, $\nu_{u,v} = 0.2$, $\tau(t = 0) = 10^{-7}$, end time $t = 4 \times 10^{-3}$, $atol = rtol = 10^{-3}$, $\Delta x = 0.01$.



(a) Contours of Profiles in 2D

(b) Graphs of profiles 3D

Figure 6.11: Precess approximation.

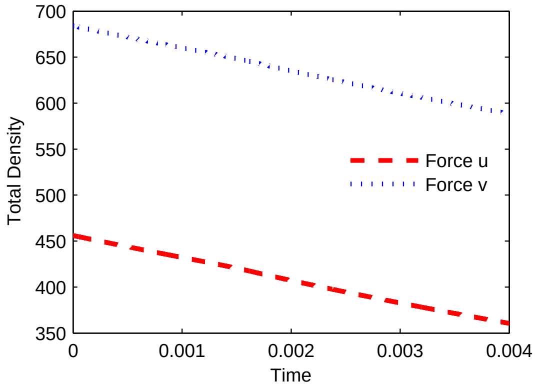
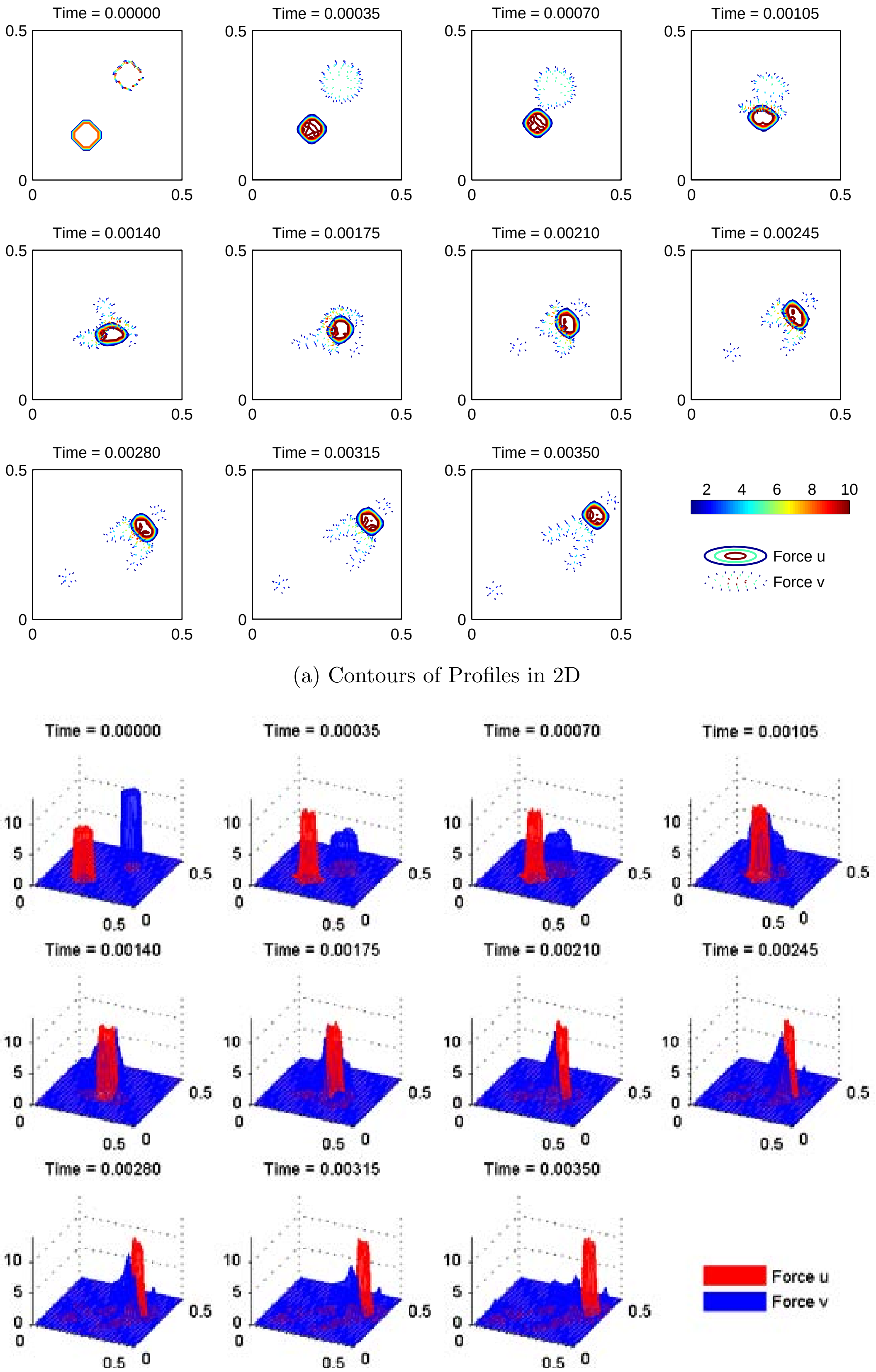


Figure 6.12: Losses for Figure 6.11.

Note that no precession behaviour is seen which corresponds to the majority of ISAAC Precess scenario simulation results. We now offset the initial positions of the forces as we did with the ISAAC scenario such that they pass to the left or right of each other to ascertain whether both clockwise and anticlockwise precession behaviour will be achieved in our continuous model.

We begin by attempting to induce anticlockwise precession. Parameters were set at $\mathrm{ID}_u = 8,\ \mathrm{ID}_v = 12,\ \rho_{u,v} = 0.05,\ \mu_u = (0.15, 0.18),\ \mu_v = (0.35, 0.32),\ \mathrm{IT}_{u,v} = 1,\ r_{a,r_{u,v}} = 5,\ D_{u,v} = 5,\ \boldsymbol{C_u} = (60, 60),\ \boldsymbol{C_v} = (-60, -60),\ A_{a_{u,v}} = 5,\ A_{r_u} = 0.5,\ A_{r_v} = 1,\ r_{S_u} = 3,\ r_{S_v} = 7,\ \Delta c_u = 10^6,\ \Delta c_v = 4,\ attack_{u,v} = -1,\ d_{u,v} = 2 \times 10^{-6},\ \beta_{u,v} = 8 \times 10^{-8},\ \nu_{u,v} = 0.2,\ \tau(t = 0) = 10^{-7}$, end time $t = 3.5 \times 10^{-3}$, $atol = rtol = 10^{-3},\ \Delta x = 0.01$.



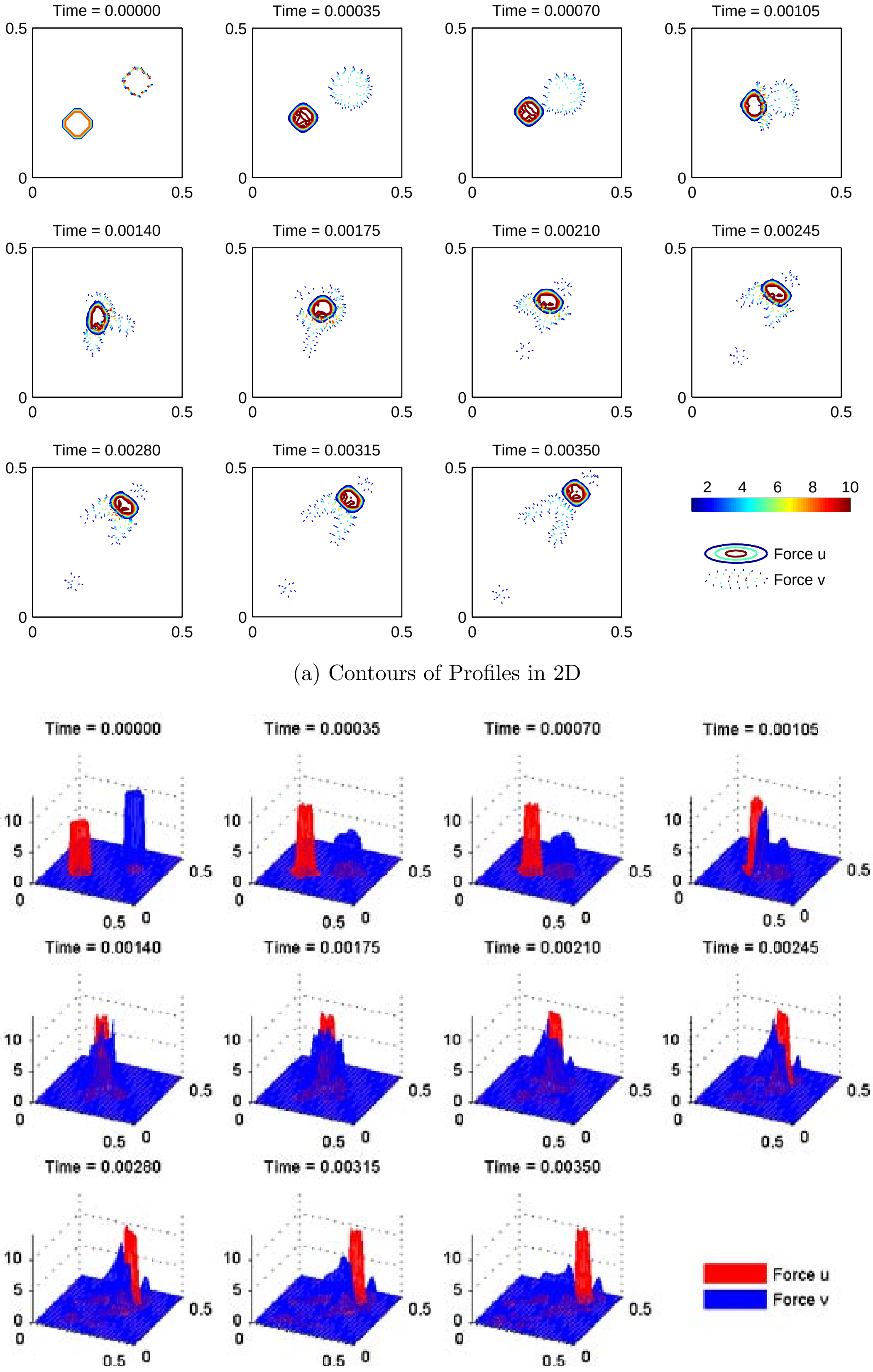
(a) Contours of Profiles in 2D

(b) Graphs of profiles 3D

Figure 6.13: Precess comparison with initial positions offset. Anticlockwise precession.

As expected, precession occurs due to the spatial asymmetry introduced into the scenario through the offset of the initial spatial distributions. The behaviour is markedly similar to that seen in Figure 6.10. Upon falling into sensor range, Force 1 is partially surrounded by Force 2 and an anti-clockwise precession occurs. Force 1 is then closely pursued by the majority of Force 2 following and a small portion directly in the path of Force 1. We now confirm our expectation that clockwise precession will occur with the initial positions offset in reverse.

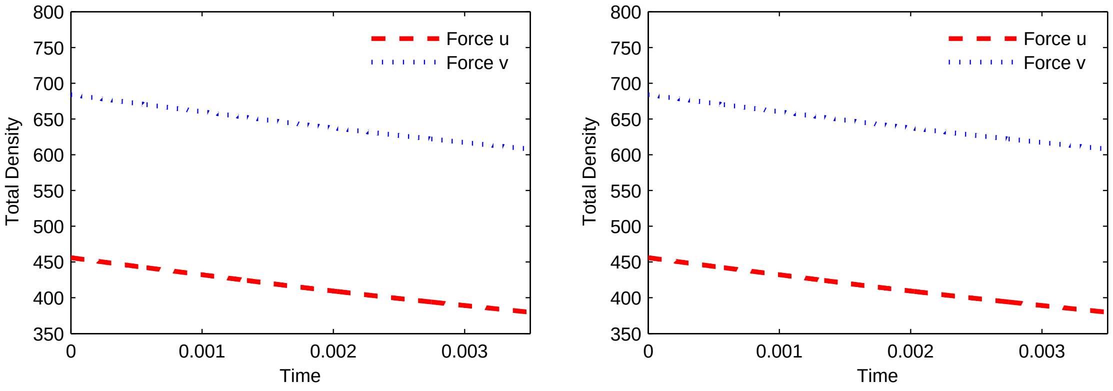
Parameters were set at $\mathrm{ID}_u = 8$, $\mathrm{ID}_v = 12$, $\rho_{u,v} = 0.05$, $\mu_u = (0.18, 0.15)$, $\mu_v = (0.32, 0.35)$, $\mathrm{IT}_{u,v} = 1$, $r_{a,r_{u,v}} = 5$, $D_{u,v} = 5$, $\boldsymbol{C_u} = (60, 60)$, $\boldsymbol{C_v} = (-60, -60)$, $A_{a_{u,v}} = 5$, $A_{r_u} = 0.5$, $A_{r_v} = 1$, $r_{S_u} = 3$, $r_{S_v} = 7$, $\Delta c_u = 10^6$, $\Delta c_v = 4$, $attack_{u,v} = -1$, $d_{u,v} = 2 \times 10^{-6}$, $\beta_{u,v} = 8 \times 10^{-8}$, $\nu_{u,v} = 0.2$, $\tau(t = 0) = 10^{-7}$, end time $t = 3.5 \times 10^{-3}$, $atol = rtol = 10^{-3}$, $\Delta x = 0.01$.



(a) Contours of Profiles in 2D

(b) Graphs of profiles 3D

Figure 6.14: Clockwise Precess Approximation, initial positions offset.



(a) Anticlockwise

(b) Clockwise

Figure 6.15: Losses for Figure 6.13 and Figure 6.14 respectively.

We propose that, similar to the Classic Fronts scenario, the precession behaviour seen in the ISAAC results (Figure 6.10) arises due to the spatial asymmetry of the forces. Depending on the spatial distribution on falling into sensor range, the precession will be either clockwise or anticlockwise, and offsetting the initial positions of the forces exacerbates this asymmetry thus increasing the frequency of precession occurring. An artificial initial spatial asymmetry in the continuous version is necessary before precession is observed.

As a result of the modification of the advection term through the addition of the *attack* variable we see minimal overlap between the forces in keeping with the ISAAC behaviour. However the positive value of *attack* for the Blue force is sufficient to give rise to pursuing of the Red force, yet the threshold value prevents any significant overlap. It is the effect of the repulsion down the gradient of highest concentration of enemy density which prevents any significant overlap.

6.4.4 Mismatch Scenario

6.4.4.1 Description

This scenario shows strikingly similar behaviour to both the Precess and Circle scenarios. The larger distribution less dense Red force has the restriction of requiring a minimum numerical advantage in order to advance upon the sensed opposition Blue force, resulting in the Blue force maintaining a separation (avoiding overlap) as this superiority is only achieved at the periphery. As the Red force does not have this restriction, it continues to advance toward the Blue flag, engaging the opposition during this advancement. It is the combination of the smaller footprint Red advancement and the separation distance which results in the seemingly “intelligent tactics” of the Blue force exploiting a superior sensor and fire range.

6.4.4.2 ISAAC Parameters and Results

Unlike the Classic Fronts scenario, the input file was not included with the software installation. However, similar to the Precession scenario, the majority of the input parameters were stated in the literature with the remainder determined through trial and error. Numerous test runs using the given parameters showed a less dense blue force advancing more rapidly than the higher density red force. Setting the *Advance* parameter to zero still produced the same behaviour, whereas its inclusion was seen as integral by Ilachinski.

Parameters used for Figure 6.16:

Parameter	Red	Blue	Parameter	Red	Blue
Squad Size	90	90	Combat	0	5
w_1	25	10	Battlefield length	100	
w_2	25	40	Battlefield width	100	
w_3	25	10	Initial Dist Centre x	10	90
w_4	25	40	Initial Dist Centre y	10	90
w_5	0	0	Size x	20	20
w_6	75	50	Size y	20	20
r_S	2	7	Flag x	1	99
r_F	1	5	Flag y	1	99
r_T	1	4	Terrain	no	
r_M	1	1	Move sampling order	random	
Prob Hit	0.002	0.002	Fratricide	no	no
Max Sim tgts	999	999	Reconstitution	no	no
Defence Measure	1	1	Terrain	no	
Cluster	0	10			

Table 6.4: ISAAC Mismatch Scenario Parameters. All other parameters are set to zero or *no*.

While Figure 6.16 in not an exact replica of the published screen shots, the equivalent behaviour is demonstrated.

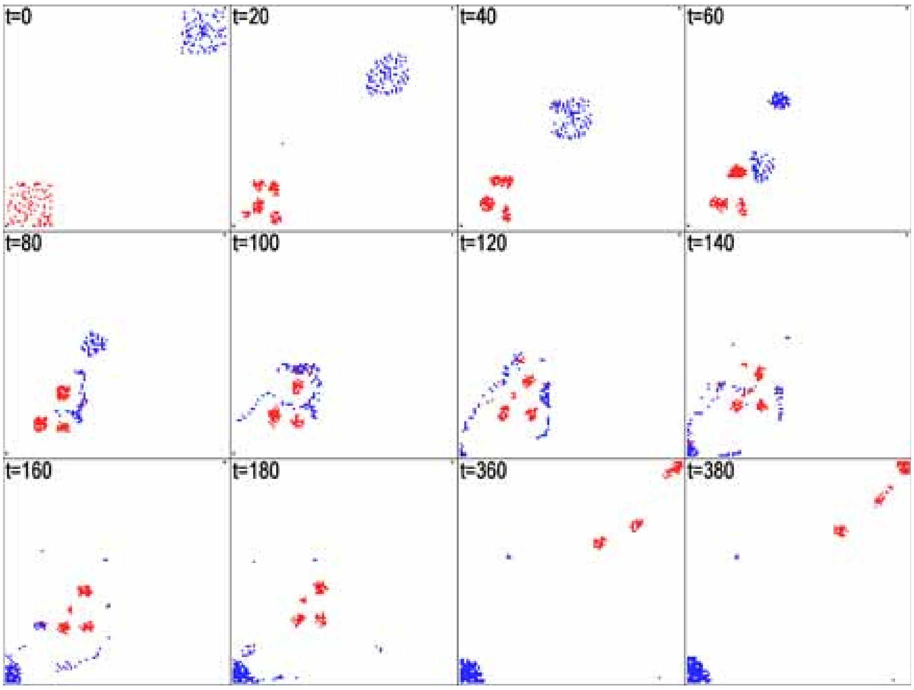


Figure 6.16: ISAAC Mismatch Scenario Screenshots

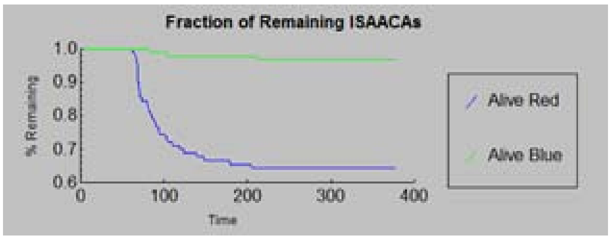
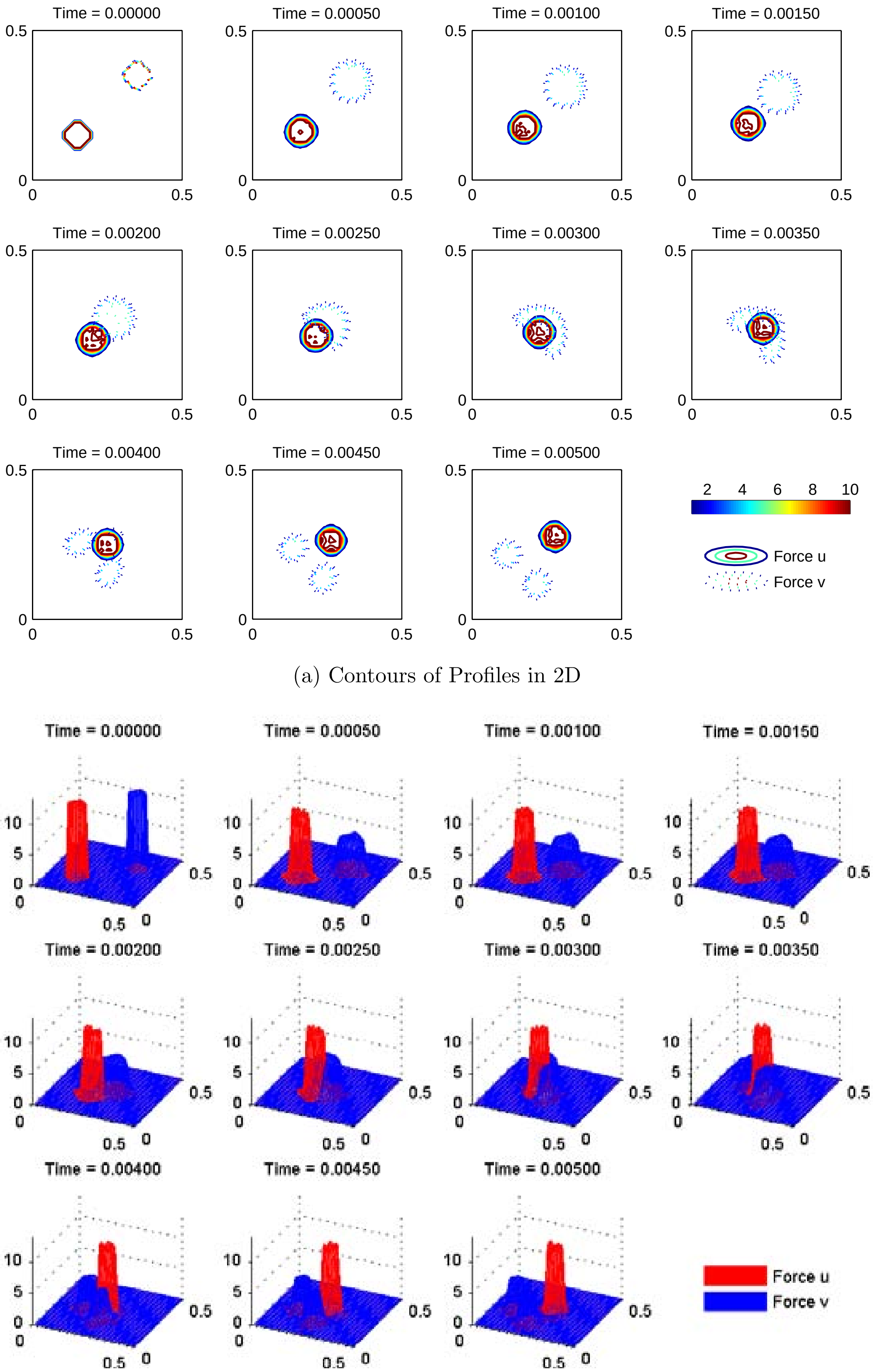


Figure 6.17: Losses for Figure 6.16

6.4.4.3 PDE Results

Using similar parameter values to the Precess scenarios and using the same form of the advection term (6.6), the following results are obtained. The only difference between the two scenarios is Aimed Fire values.

Parameters were set at $\mathrm{ID}_{u,v} = 10$, $\rho_{u,v} = 0.05$, $\mu_u = (0.15, 0.15)$, $\mu_v = (0.35, 0.35)$, $\mathrm{IT}_{u,v} = 0.5$, $r_{a,r_{u,v}} = 5$, $D_{u,v} = 5$, $\boldsymbol{C_u} = (40, 40)$, $\boldsymbol{C_v} = (-60, -60)$, $A_{a_{u,v}} = 5$, $A_{r_u} = 0.5$, $A_{r_v} = 1$, $r_{S_u} = 0$, $r_{S_v} = 3$, $\Delta c_u = 0$, $\Delta c_v = 2000$, $attack_u = 0$, $attack_v = -1$, $d_u = 10^{-5}$, $d_v = 10^{-5}$, $\tau(t = 0) = 10^{-7}$, end time $t = 5 \times 10^{-3}$, $atol = rtol = 10^{-3}$, $\Delta x = 0.01$.



(a) Contours of Profiles in 2D

(b) Graphs of profiles 3D

Figure 6.18: Mismatch approximation with Aimed Fire Only.

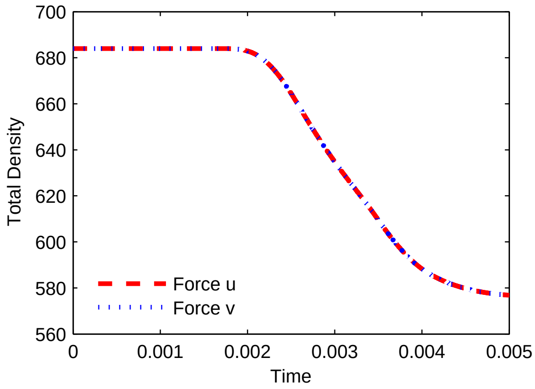


Figure 6.19: Losses for Figure 6.18.

Similarly to the ISAAC results shown in Figure 6.16, the denser Red Force is surrounded by the more aggressive Blue Force. While the repulsion due to force inferiority maintains a degree of separation between the two forces, some overlap is present at the edges. Once the Red Force is surrounded, overall movement slows to almost a halt. Density is gradually lost due to attrition and there is a slight imbalance in the Blue Force's distribution around the Red Force. The slow advance of Red and imbalance in the Blue Force distribution results in an eventual breakaway of Red. Blue is divided into roughly two separate groups, one on each side of the Red Force similar to the ISAAC scenario.

6.4.5 Circle Scenario

6.4.5.1 Description

This scenario shows strikingly similar behaviour to the Precess scenario. Both forces form dense formations with Blue slightly denser due to the higher cluster variable. The more aggressive Red force envelops the Blue force entirely upon coming into sensor range and it is at this point, $time = 140$, that the published snapshots stop. This can give the impression that this formation remains static or stable after this time. When running the simulation past this time step, the similarity with the Precess scenario from $time = 120$ onwards becomes apparent. Blue forces continue to move towards the Blue flag and Red forces begin to concentrate on the opposite side (Figure 6.20, $time = 200$). For the remainder of the simulation, Red forces pursue the Blue force toward the Blue flag. In this particular time series, a smaller section of the Blue force is separated and pursued by the Red force ($time = 220$) in the same manner as the larger section.

6.4.5.2 ISAAC Parameters and Results

Unlike the Classic Fronts scenario and similar to the Precession scenario, no input file was provided with the software installation, only a series of snapshots with a partial list of parameters and corresponding values. Initial distribution position and size values were not provided. Numerous test runs using the given parameters showed a less dense blue force advancing more rapidly than the higher density red force. Setting the *Advance* and *Minimum Distance* parameters to zero still produced the same behaviour.

Parameters used for Figure 6.20:

Parameter	Red	Blue	Parameter	Red	Blue
Squad Size	200	200	Combat	-7	5
w_1	10	25	Battlefield length	100	
w_2	50	25	Battlefield width	100	
w_3	0	75	Initial Dist Centre x	10	90
w_4	100	25	Initial Dist Centre y	10	90
w_5	0	0	Size x	20	20
w_6	25	75	Size y	20	20
r_S	5	5	Flag x	1	99
r_F	3	3	Flag y	1	99
r_T	3	3	Terrain	no	
r_M	1	1	Move sampling order	random	
Prob Hit	0.001	0.001	Fratricide	no	no
Max Sim tgts	999	999	Reconstitution	no	no
Defence Measure	1	1	Terrain	no	
Cluster	3	15			

Table 6.5: ISAAC Circle Scenario Parameters. All other parameters are set to zero or *no*.

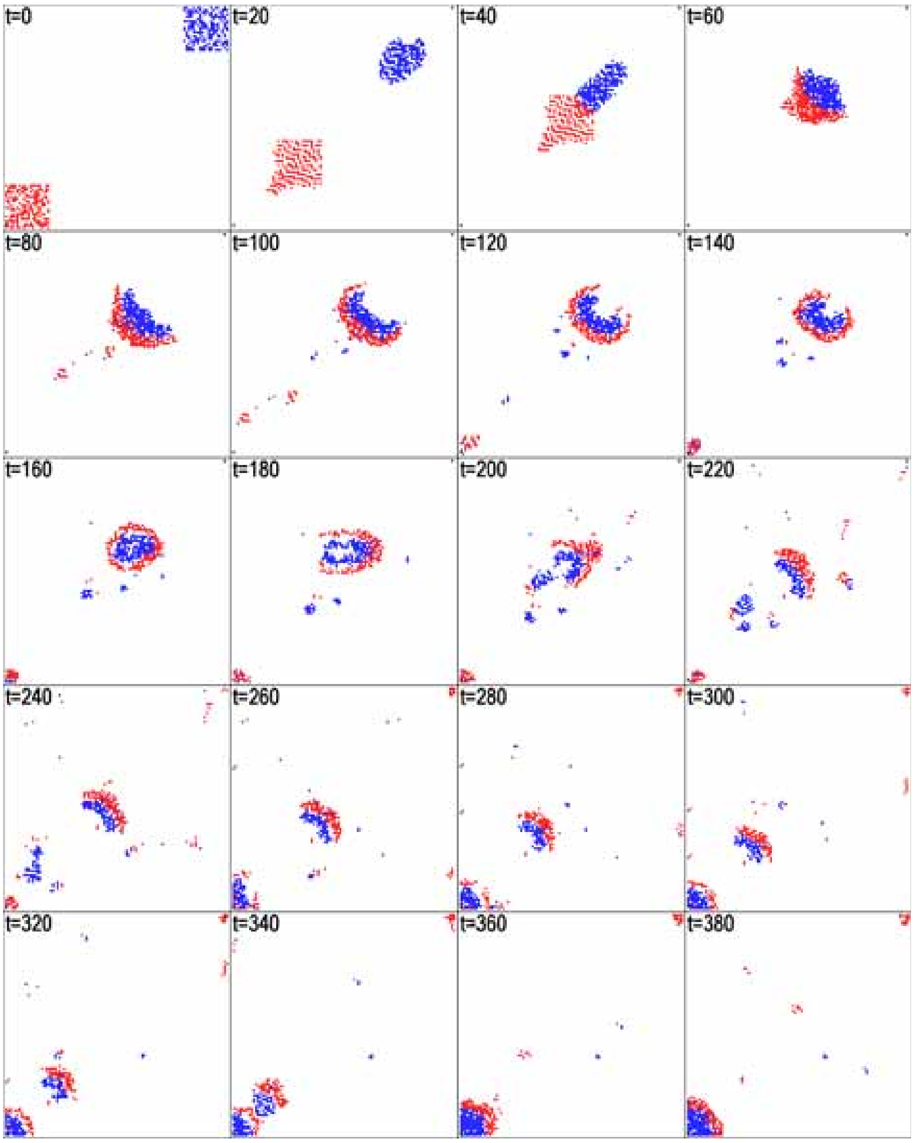


Figure 6.20: ISAAC Circle Scenario Screenshots

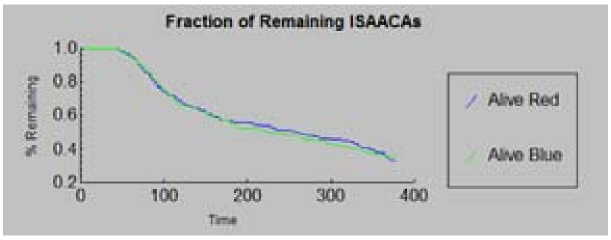
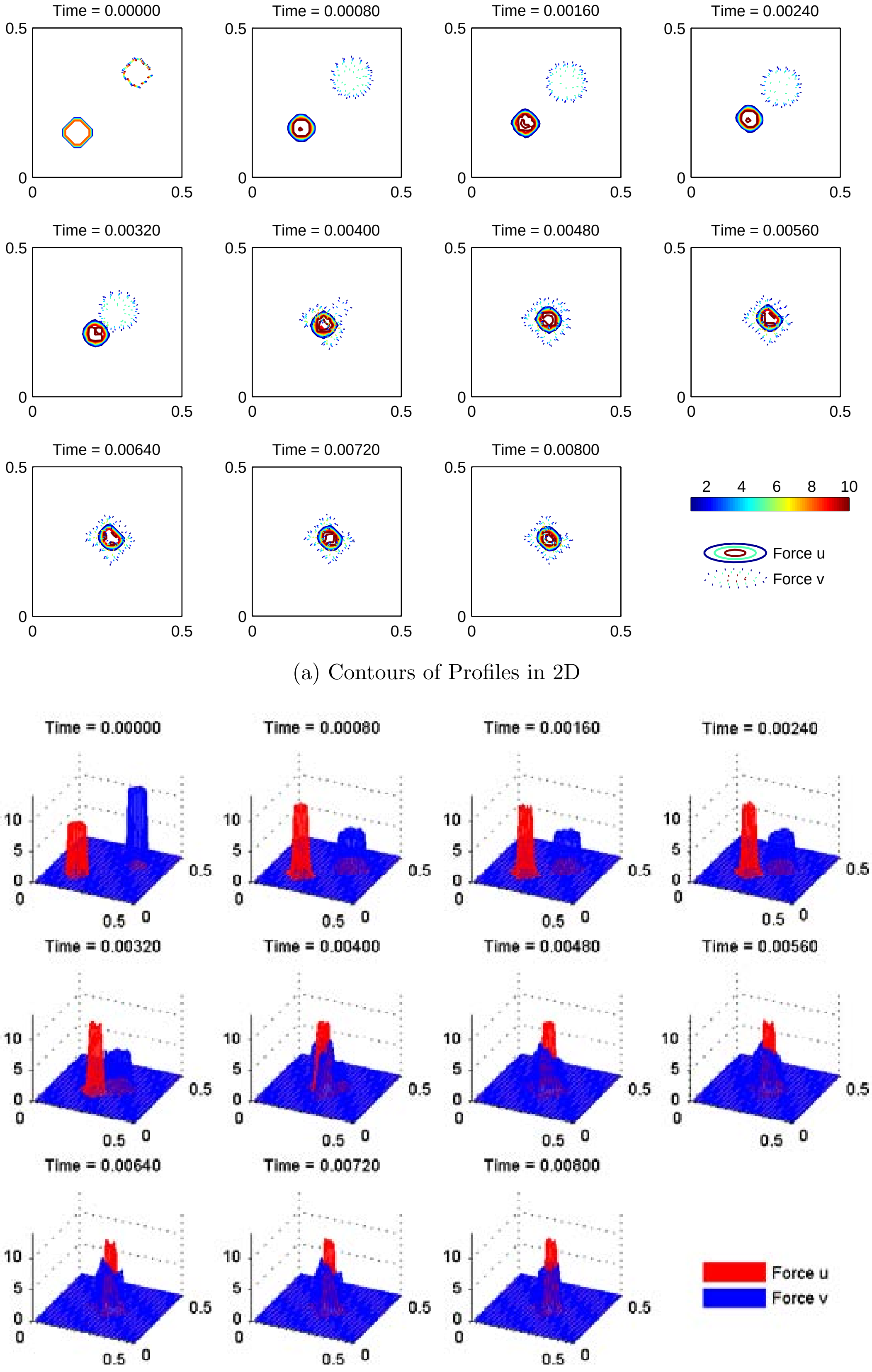


Figure 6.21: Losses for Figure 6.20.

6.4.5.3 PDE Results

Using similar parameter values to the Precess scenarios and using the same form of the advection term (6.6), the following results are obtained. The only difference between the two scenarios is Aimed Fire values.

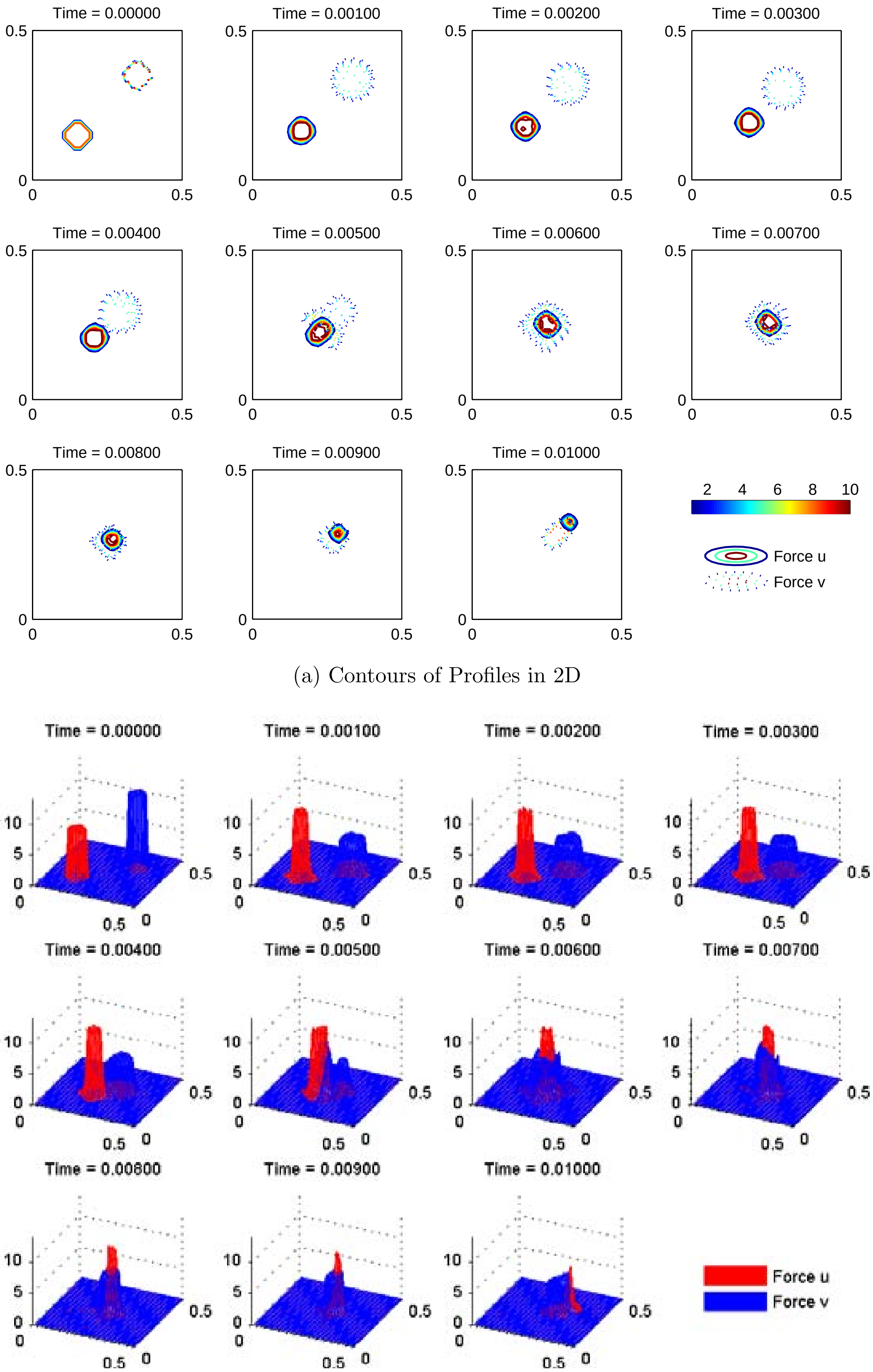
Parameters were set at $\mathrm{ID}_u = 8$, $\mathrm{ID}_v = 12$, $\rho_{u,v} = 0.05$, $\mu_u = (0.15, 0.15)$, $\mu_v = (0.35, 0.35)$, $\mathrm{IT}_{u,v} = 1$, $r_{a,r_{u,v}} = 5$, $D_{u,v} = 5$, $\boldsymbol{C_u} = (20, 20)$, $\boldsymbol{C_v} = (-20, -20)$, $A_{a_{u,v}} = 5$, $A_{r_u} = 0.5$, $A_{r_v} = 1$, $r_{S_u} = 3$, $r_{S_v} = 4$, $\Delta c_{u,v} = 18$, $attack_{u,v} = -1$, $d_{u,v} = 2 \times 10^{-6}$, $\tau(t = 0) = 10^{-7}$, end time $t = 8 \times 10^{-3}$, $atol = rtol = 10^{-3}$, $\Delta x = 0.01$. For Figure 6.23 $d_u = 10^{-4}$, $d_v = 10^{-4}$.



(a) Contours of Profiles in 2D

(b) Graphs of profiles 3D

Figure 6.22: Circle approximation with Aimed Fire Only.



(a) Contours of Profiles in 2D

(b) Graphs of profiles 3D

Figure 6.23: Circle approximation with with differing values of Aimed Fire to Figure 6.22.

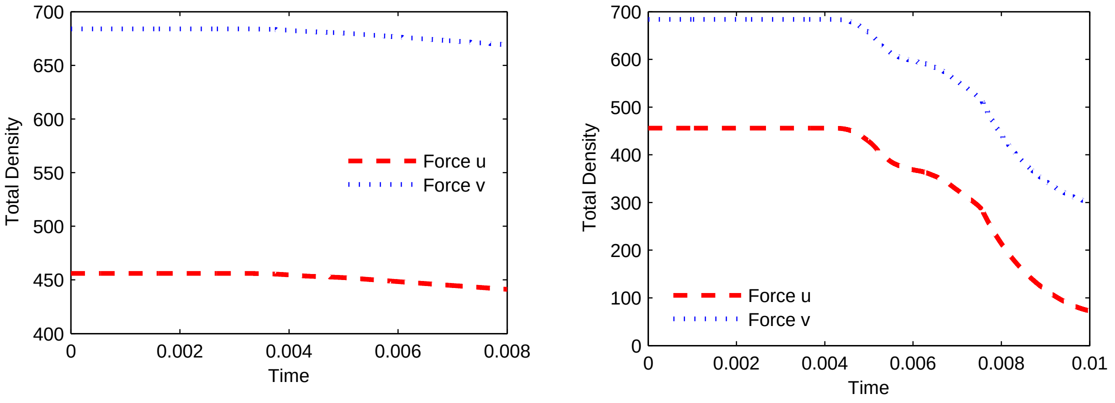


Figure 6.24: Losses for Figure 6.22 and Figure 6.23 respectively

Similarly to the ISAAC results shown in Figure 6.20, in Figure 6.22 the denser Force 1 is surrounded by the more aggressive Force 2. While the repulsion due to force inferiority maintains a degree of separation between the two forces, some overlap is present at the edges. Once Force 1 is surrounded, overall movement slows to almost a halt. Density is gradually lost due to attrition and there is a slight imbalance in Force 2's distribution around Force 1. The slow advance of Force 1 and imbalance in Force 2 results in an eventual breakaway of Force 1. The higher attrition rate used in Figure 6.23 shows that this almost stationary phase has a much shorter duration as attrition speeds the generation of an imbalance in the surrounding force's density.

Note the nonlinear form of density loss shown in Figure 6.24. Lauren states that casualty rates are uneven on a turbulent nonlinear battlefield. This is held as an important distinction between complex adaptive and convectional combat models. Here we have shown that a conventional model is indeed capable of producing intermittent density losses.

6.4.6 Sensor Scenario

6.4.6.1 Description

This set of scenarios shows the effect of increasing sensor range while keeping all other parameters fixed. An almost complete lack of specified parameters made recreating the published snapshots extremely difficult and the stated SENSOR_X.out files were not included in any versions of the ISAAC/EINSTein installations. Only force sizes, sensor ranges, fire range, single-shot probabilities and Combat thresholds were specified.

Specifically, this suite of four scenarios varies only in the sensor range of the doubly larger Red force. As this value increases to equal that of the Blue force and then surpasses that value, a noticeable change in the interaction dynamics is observed. At sensor ranges equal to or lower than the Blue force, Red penetrates the Blue force which then disperses and forms small clusters at the periphery of the Red force where engagement occurs. Throughout these scenarios, the Red force advances at a relatively constant velocity to the Blue goal as seen in Red Sensor=3 and Red Sensor=5 in Figure 6.25. As the Red sensor range is increased beyond parity, Red fails to penetrate Blue and the inner/outer force roles are reversed. This encircling behaviour is similar to the Precess and Circle scenarios where the Combat threshold produces a repulsive velocity down the gradient of greatest enemy density within sensor range. As the Blue force maintains a large central formation of troops, this repulsion results in the Red force partially surrounding the Blue force. As the sensor range increases, this effect becomes more pronounced and the Red force is not as effective at reaching the Blue goal due to these combat effects dominating the goal term in the penalty function.

This scenario is used by Ilachinski to demonstrate that an increase in sensor range does not produce a monotonic increase in overall effectiveness. Effectiveness here is described by two features; (i) the Red force's effectiveness at establishing a presence near the Blue flag, and (ii) defending the Blue advancement to the Red flag.

6.4.6.2 ISAAC Parameters and Results

Parameters used for Figure 6.25:

Parameter	Red	Blue	Parameter	Red	Blue
Squad Size	100	50	Advance		20
w_1	10	10	Combat	-4	0
w_2	40	30	Battlefield length	80	
w_3	10	10	Battlefield width	80	
w_4	40	30	Initial Dist Centre x	20	60
w_5	0	0	Initial Dist Centre y	40	40
w_6	30	30	Size x	20	20
r_S	7	5	Size y	10	10
r_F	4	4	Flag x	1	80
r_T	3	3	Flag y	40	40
r_M	1	1	Terrain	no	
Prob Hit	0.005	0.005	Move sampling order	random	
Max Sim tgts	3	3	Fratricide	no	no
Defence Measure	1	1	Reconstitution	no	no
Cluster	10	15	Terrain	no	

Table 6.6: ISAAC Sensor Scenario Parameters. All other parameters are set to zero or *no*.

As mentioned, due to the few parameters specified, the method of trial-and-error was used to determine the remaining values. The screenshots presented in Figure 6.25 display the same main features as published.

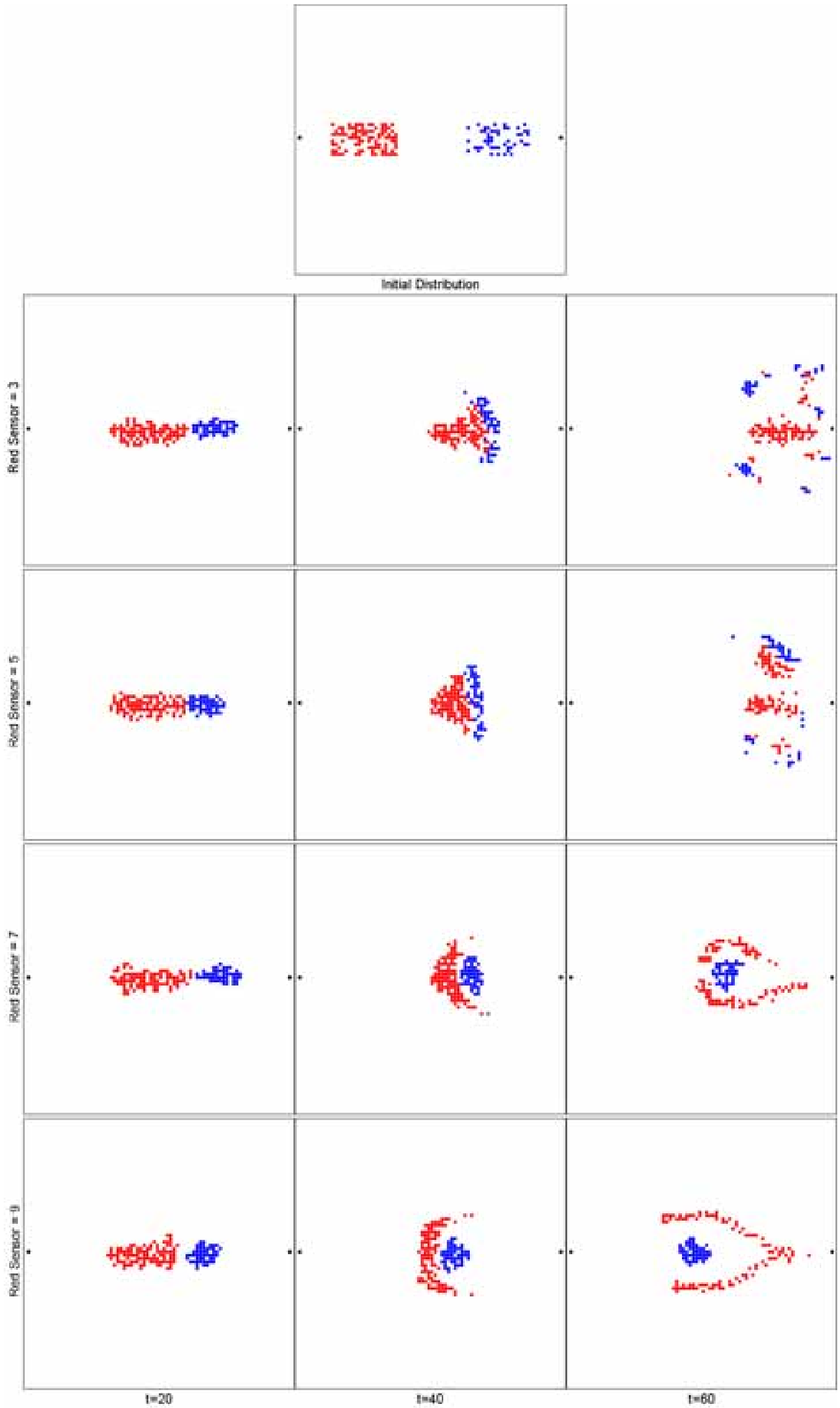


Figure 6.25: ISAAC Circle Scenario Screenshots

6.4.6.3 PDE Results

Parameters were set at $\text{ID}_{u,v} = 10$, $\rho_{u,v} = 0.05$, $\mu_u = (0.15, 0.15)$, $\mu_v = (0.35, 0.35)$, $\text{IT}_{u,v} = 0.5$, $r_{a,r_{u,v}} = 5$, $D_{u,v} = 5$, $\boldsymbol{C_u} = (20, 0)$, $\boldsymbol{C_v} = (-20, 0)$, $A_{a_{u,v}} = 5$, $A_{r_{u,v}} = 0.5$, $r_{S_{u,v}} = 3$, $\Delta c_{u,v} = 100$, $attack_{u,v} = -1$, $d_u = 10^{-5}$, $d_v = 10^{-5}$, $\tau(t = 0) = 10^{-7}$, end time $t = 5 \times 10^{-3}$, $atol = rtol = 10^{-3}$, $\Delta x = 0.01$.

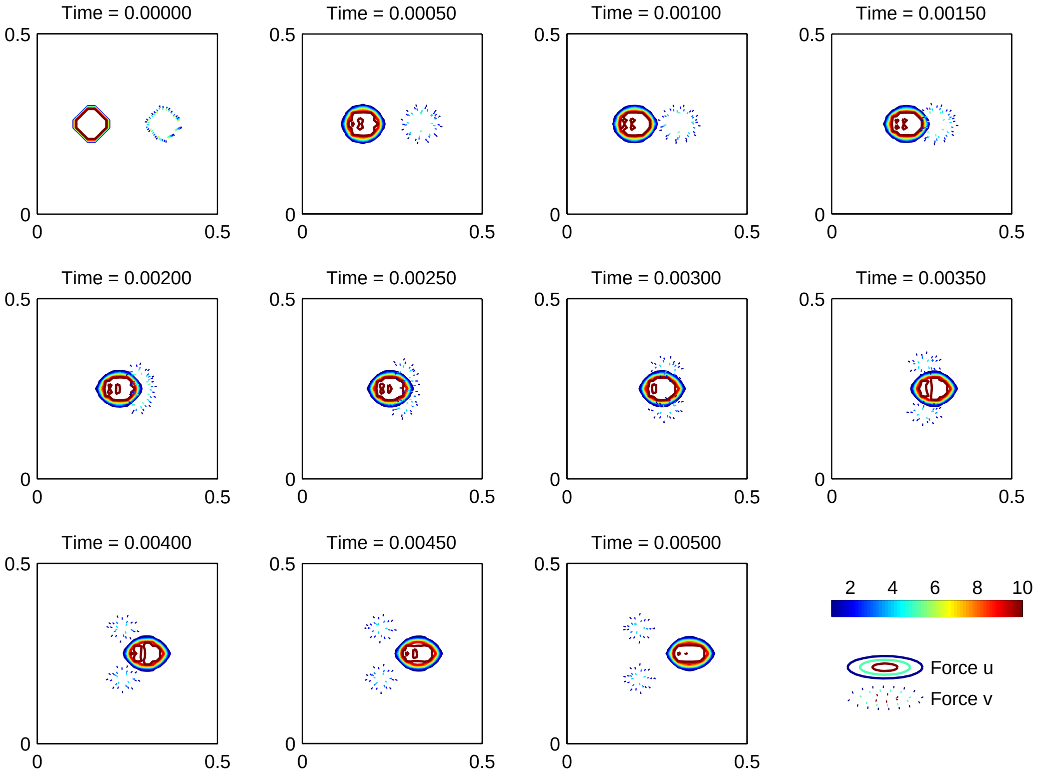


Figure 6.26: Sensor approximation with Aimed Fire Only, Sensor range = 1.

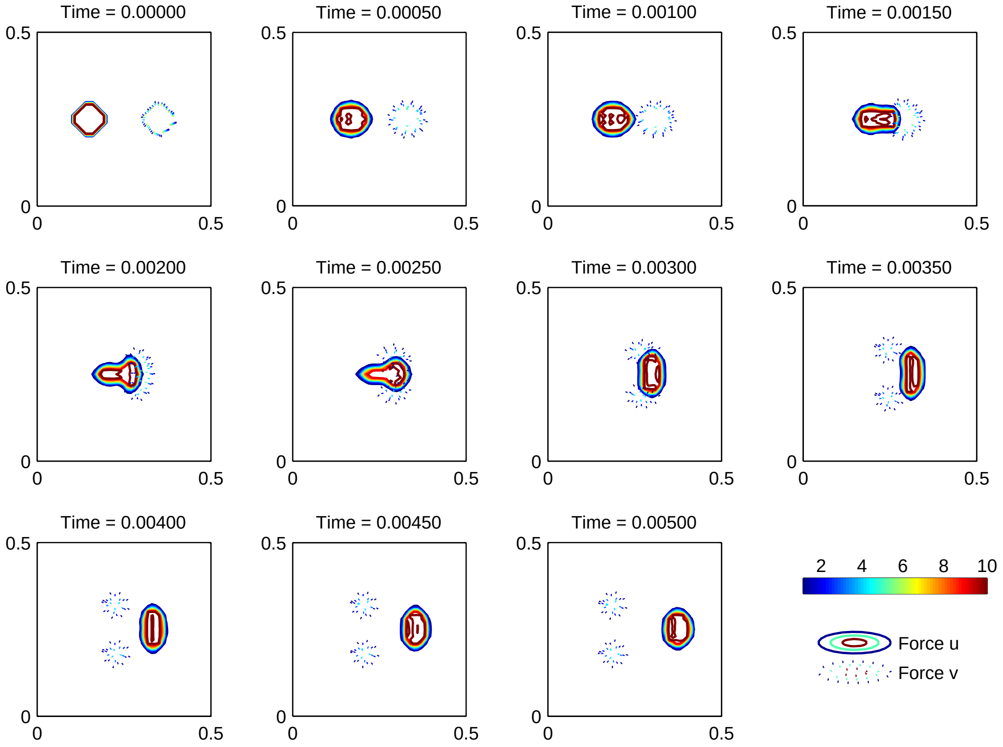


Figure 6.27: Sensor approximation with Aimed Fire Only, Sensor range = 3.

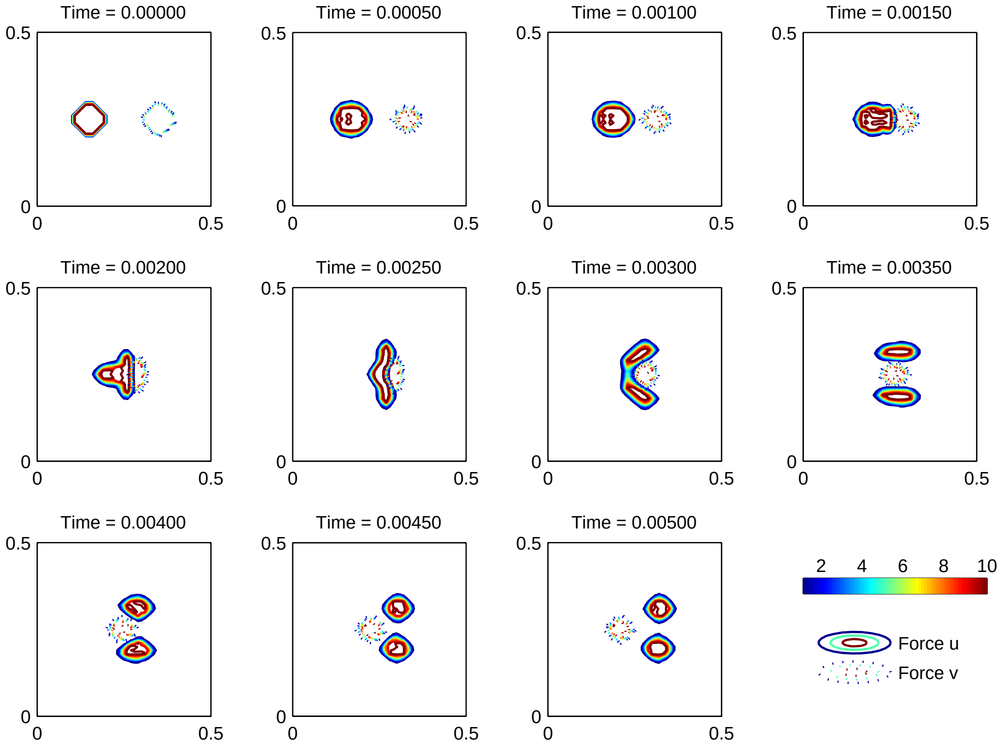


Figure 6.28: Sensor approximation with Aimed Fire Only, Sensor range = 5.

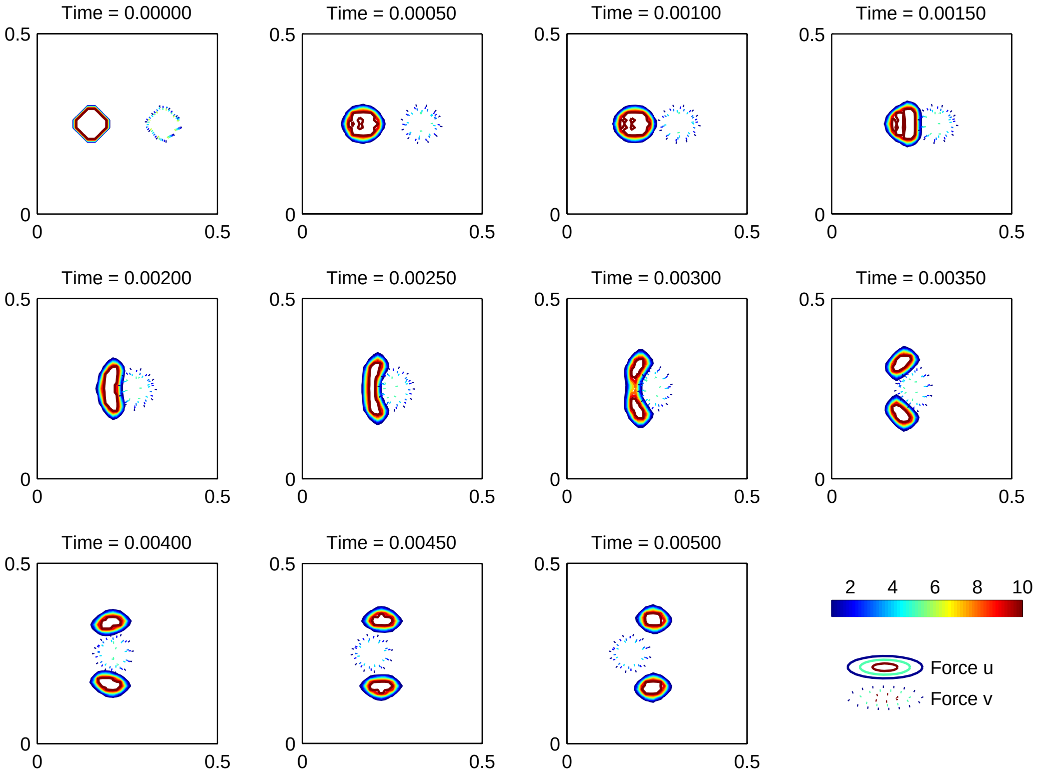


Figure 6.29: Sensor approximation with Aimed Fire Only, Sensor range = 7.

Our series of four scenarios in Figures 6.26, 6.27, 6.28 and 6.29 demonstrates how an increasing sensor range r_c increases the effect of the convolution term in 6.6. That is, an increasing convolution term results in a stronger repulsion down the gradient of greatest enemy density if the threshold density Δc is not exceeded. This is clearly demonstrated in both discrete and continuous results.

6.5 Discussion

As the comparisons between the ISAAC and our continuous scenarios indicate, there lies an inherent danger in relying on one realisation of a discrete scenario to drive analysis or to be held as a valid representation of the output with the given input parameters. The Precess scenario is a prime example of this as the majority of simulations displayed behaviour more similar to the Circle scenario. Precession behaviour was absent in the majority of ISAAC simulations run. Implementing a method of ascertaining the reliability of ISAAC results by providing some form of confidence limit as discussed in [11] is recommended for those interested in using ISAAC as a research tool.

However, we must note that the spatial asymmetry produced through ISAAC's inherent randomness can result in behaviour not seen in our continuous counterpart on first try. By artificially inducing asymmetry through careful positioning of the initial distributions of the continuous force density profiles, these behaviours were replicated. In this way, our PDEs act to establish the mean behaviour and we retain control over how this approximation of stochasticity is introduced.

Tactics such as forward advance, frontal attack, local clustering, penetration, retreat, containment, flanking manoeuvres, defensive posturing and encirclement, defined as *self-organised emergent behaviours* and part of the *rich spectrum of ISAACA behaviours* are all demonstrated in our *continuous* model. We have shown that a Lanchester-type interaction model extended to include aggregation spatial dynamics also displays this rich spectrum.

The aim of this thesis was not to develop a continuous model directly from the discrete ISAAC model, rather to extend the work by Protopopescu *et al.* and to introduce the essential nonlocal features of attraction to "like" members and some form of response to "unlike" members as is common in CA wargames. As the terms in the ISAAC penalty function are essentially discrete convolution terms, the swarming integro-differential equation with nonlocal interaction as developed by Mogilner and Edelstein-Keshet was an excellent position to start from as this model neatly displayed the desired characteristics of soldier movement with physically meaningful terms. The flexibility in using Mogilner and Edelstein-Keshet's work as a starting point has enabled three important things; the formation of soldier profiles with constant interior density and steep edges with the ability to navigate the battlespace while maintaining this formation; the reproduction of ISAAC-type behaviour through the addition of other interaction terms; and more importantly, the easy introduction and exploration of new tactics such as the Density Response tactic which will be shown in Chapter 8.

Chapter 7

ISAAC Comparison to PDEs

The comparisons made in the previous chapter highlight the dependency of the evolution of the discrete ISAAC equations on the parameters as given in [18]. We now investigate the dynamics of two equal forces for a range of simple scenarios using our PDE model. Initial position and *attack* variable are varied with all other parameters held constant and we seek to reproduce these results in ISAAC as closely as possible. This is the opposite of the previous chapter where continuous counterparts were derived to match given ISAAC scenarios.

We use the form of our equations as developed in Chapter 6 that utilises the *attack* variable:

$$\frac{\partial u}{\partial t} = \nabla\cdot(\boldsymbol{D_u}(u)\nabla u)+\nabla\cdot\{u(\boldsymbol{C_u}u+A_a(K_a{*}u)-A_r u(K_r{*}u))\}+u\,(k_u * v)+d_u v \quad (7.1)$$

$$\frac{\partial v}{\partial t} = \nabla\cdot(\boldsymbol{D_v}(v)\nabla v)+\nabla\cdot\{v(\boldsymbol{C_v}v+A_a(K_a{*}v)-A_r v(K_r{*}v))\}+v\,(k_v * u)+d_v u \quad (7.2)$$

where for Force u

$$\boldsymbol{C} = \begin{cases} C + \int_{x-r_C}^{x+r_C}\int_{y-r_C}^{y+r_C} v \; dxdy & N_u - N_v \geq \Delta c \\ \\ C + attack \times \int_{x-r_C}^{x+r_C}\int_{y-r_C}^{y+r_C} v \; dxdy & N_u - N_v < \Delta c \end{cases} \quad (7.3)$$

and similary for Force v.

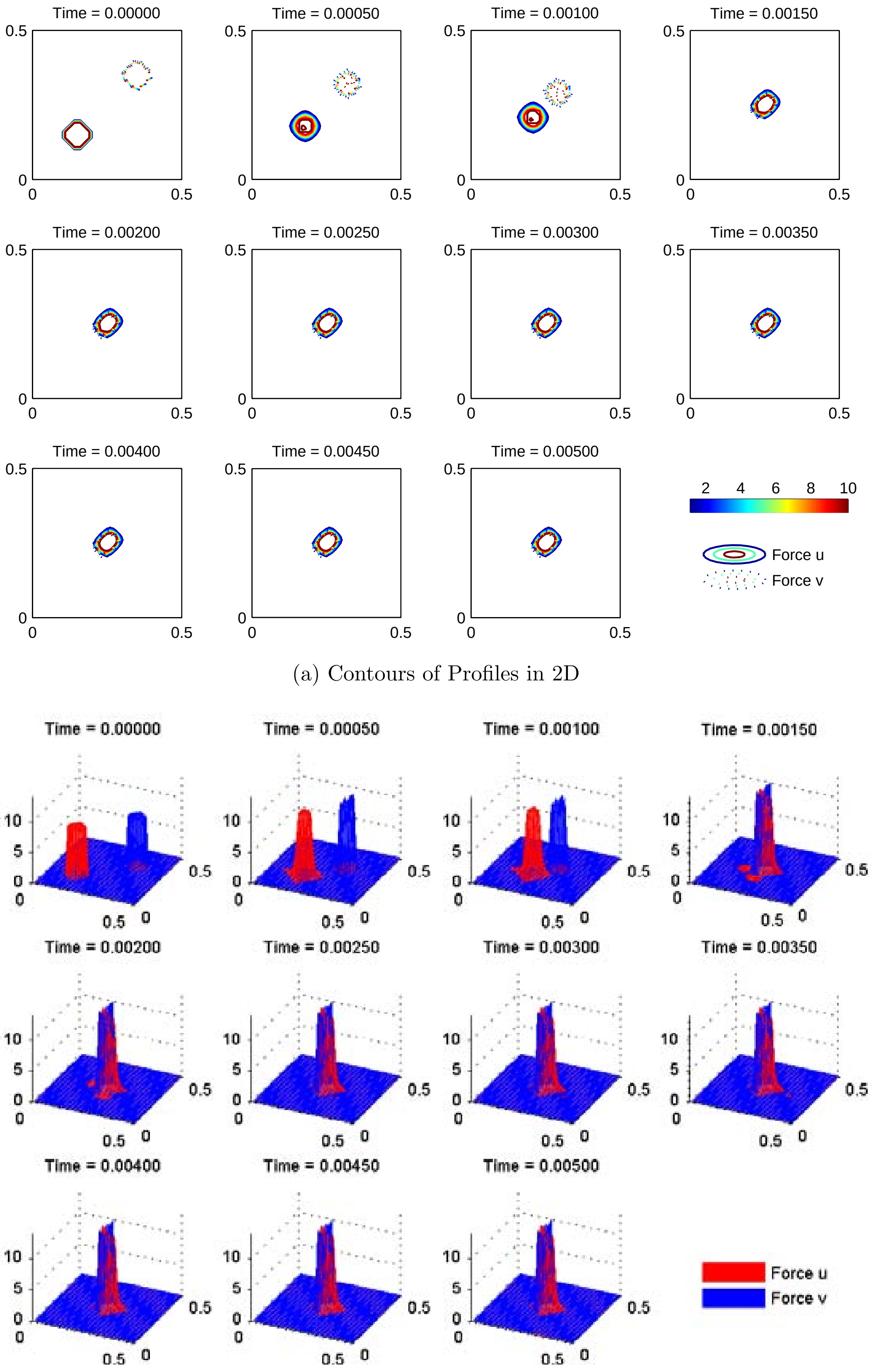
Firstly the effect of varying the simplest parameter, *attack*, on the dynamics is investigated. Following the observations from the previous chapter of the effect of spatial asymmetry, we again introduce this through a slight spatial offset in the forces initial distribution and thus velocity vectors.

Following the previous chapter, distributions of initial densities will be of the form:

$$w_i(x,y,0) = \begin{cases} \text{ID} & (\sqrt{\frac{\rho}{\rho\sqrt{2\pi}}} * e^{\frac{-|c-\mu|}{2\rho^2}}) > \text{IT} \\ 0 & \text{otherwise} \end{cases} \qquad i = u, v \tag{7.4}$$

7.1 Mutual Attack

Now setting the initial force sizes, attraction, repulsion, firing and attack variables equal, we investigate the effect of initial conditions and the *attack* variable. Firstly, *attack* is set to 1 so that each force will pursue the enemy once sensed, regardless of numerical superiority. As expected, once the enemy is sensed, both forces rapidly co-locate and remain so while slowly attritting. If the simulation time is set significantly longer or the fire coefficients set higher, both forces total densities will reduce to zero as expected. Parameters were set at $\text{ID}_{u,v} = 10$, $\rho_{u,v} = 0.05$, $\mu_u = (0.15, 0.15)$, $\mu_v = (0.35, 0.35)$, $\text{IT}_{u,v} = 0.5$, $r_{a,r_{u,v}} = 5$, $D_{u,v} = 5$, $\boldsymbol{C_u} = (60, 60)$, $\boldsymbol{C_v} = (-60, -60)$, $A_{a_{u,v}} = 5$, $A_{r_{u,v}} = 0.5$, $r_{C_{u,v}} = 5$, $\Delta c = 20$, $attack_{u,v} = -1$, $d_{u,v} = 2 \times 10^{-6}, \beta_{u,v} = 8 \times 10^{-8}$, $\nu_{u,v} = 0.2$, $\tau(t = 0) = 10^{-7}$, end time $t = 5 \times 10^{-3}$, $atol = rtol = 10^{-3}$, $\Delta x = 0.01$.



(a) Contours of Profiles in 2D

(b) Graphs of profiles 3D

Figure 7.1: Equal Forces, Both Attacking

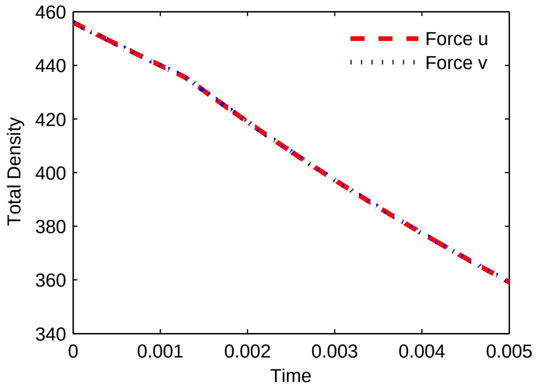


Figure 7.2: Losses for Figure 7.1

7.1.1 ISAAC Approximation

Now an approximation of this scenario is made using ISAAC. As using the *default* lattice density of one troop per cell to mimic the repulsion term in (2.6) and (2.7) is somewhat inelegant, the Cluster meta-personality parameter is used in combination with w_1 and w_3 to mimic the attraction and repulsion terms (see Figure 6.3). If the default density method is used, mingling of forces is hindered as a greater percentage of lattice sites within the force footprint are occupied. Allowing for a sparser density will permit greater mingling, thereby imitating the ability of the continuous model to allow both forces to simultaneously occupy the same spatial region. By setting a high negative value for the Combat parameter, both forces are strongly attracted up the density gradient of opposition forces within the given sensor range.

Parameter	Red	Blue	Parameter	Red	Blue
Squad Size	200	200	Combat	-20	-20
w_1	10	10	Battlefield length	100	
w_2	50	50	Battlefield width	100	
w_3	10	10	Initial Dist Centre x	10	90
w_4	50	50	Initial Dist Centre y	50	50
w_5	0	0	Size x	20	20
w_6	20	20	Size y	20	20
r_S	5	5	Flag x	1	100
r_F	2	2	Flag y	1	100
r_T	2	2	Terrain	no	
w_M	2	2	Move sampling order	random	
Prob Hit	0.002	0.002	Fratricide	no	no
Max Sim tgts	5	5	Reconstitution	no	no
Defence Measure	1	1	Terrain	no	
Cluster	10	10			

Table 7.1: ISAAC Parameters Equal Forces Mutual Attack. All other parameters are set to zero or *no*.

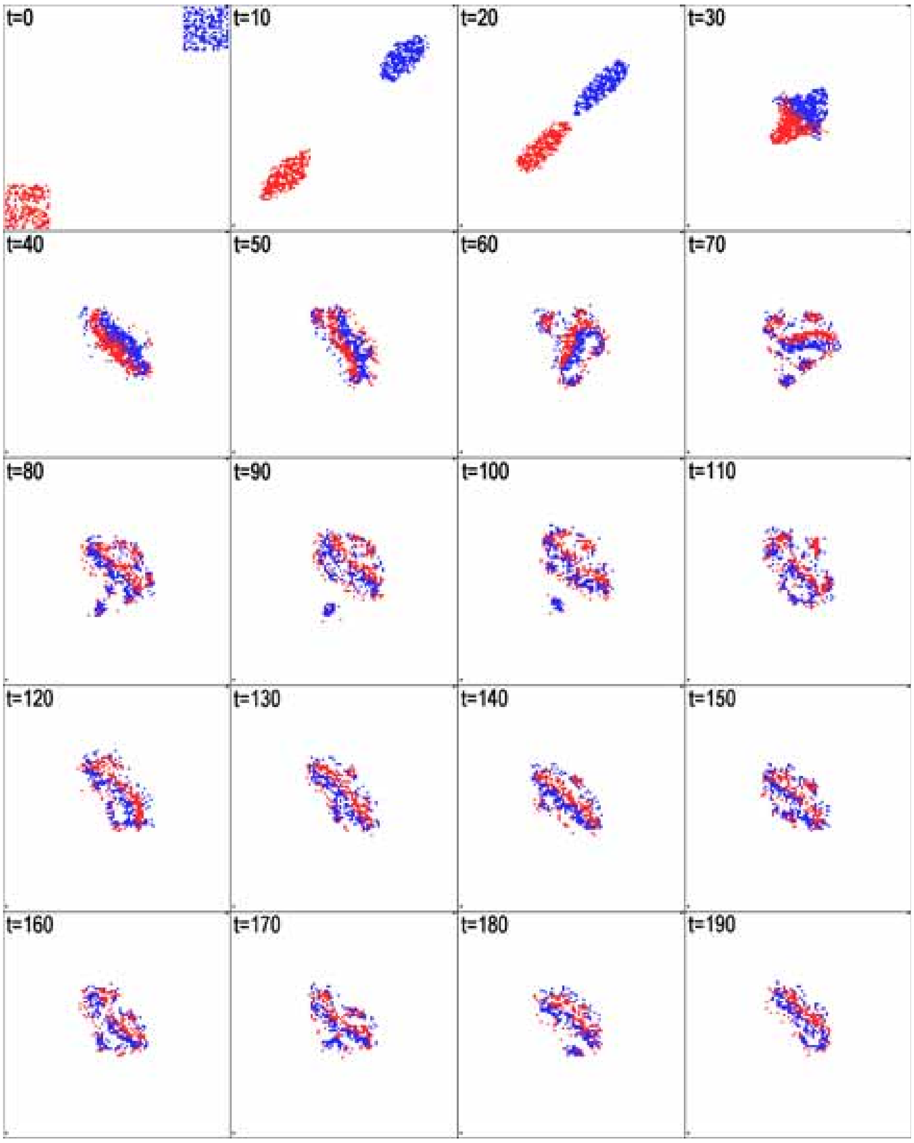


Figure 7.3: Screenshots of ISAAC Approximation to Equal Forces Mutual Attack Scenario

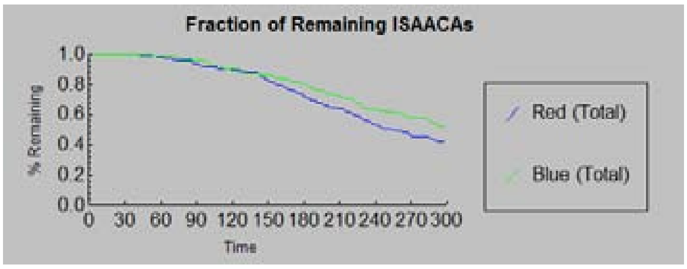
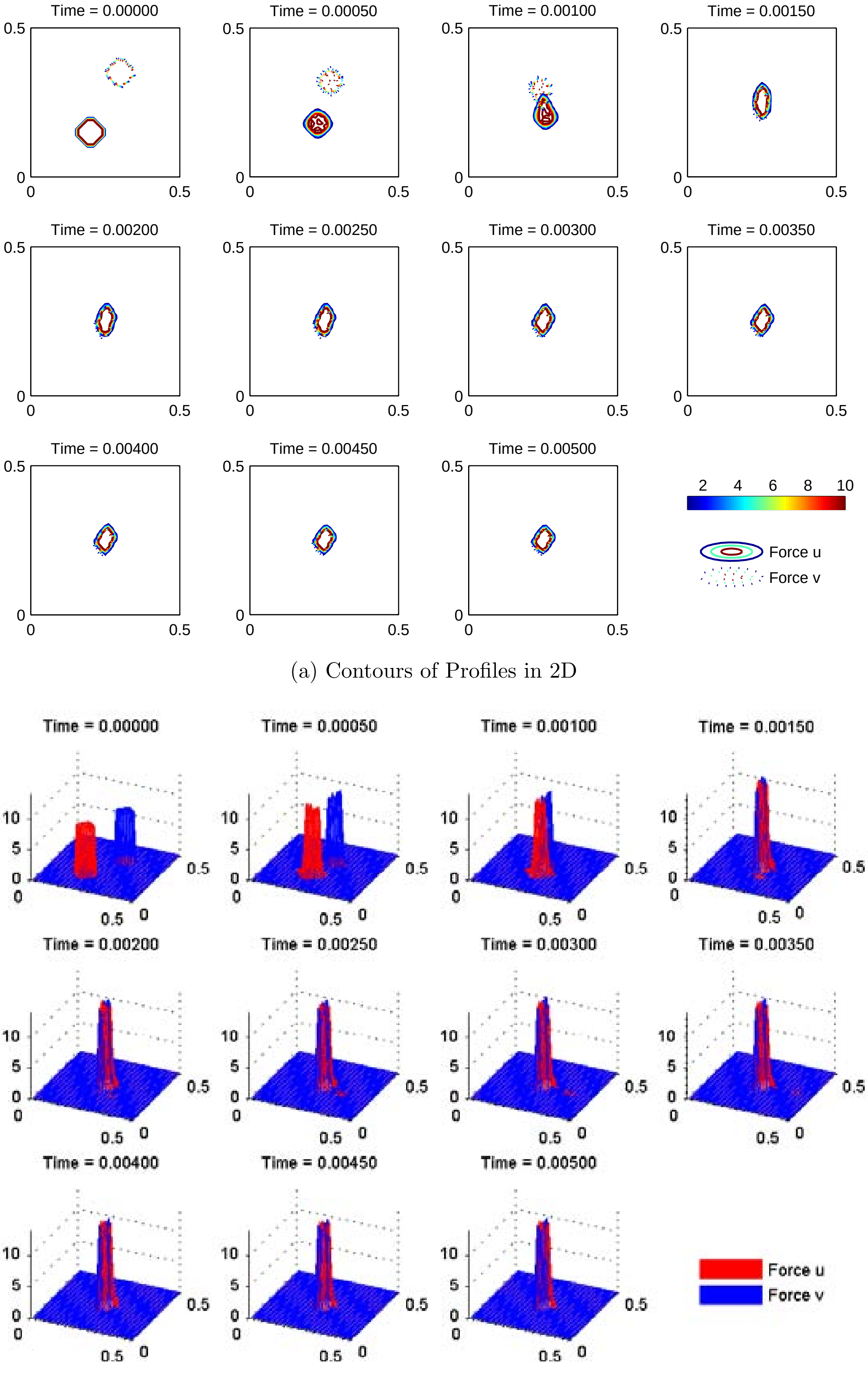


Figure 7.4: Losses for Figure 7.3

As expected, upon each force falling into sensor range of the other, both forces engage while located in the central region of the domain similar to Figure 7.1. There is some movement away from this location as expected due to the stochasticity present. Attrition in using both methods shows a gradual decline in force numbers (Figures 7.2 and 7.4).

7.2 Mutual Attack, Offset Initial Position

Offsetting the initial positions of the forces enables one instance of asymmetry to be examined, however we expect a similar overall behaviour to Figure 7.1. That is, once the convolution term in 7.3 becomes nonzero, the forces will co-locate, remain stationary and continue to inflict attrition on the opposite force until either force density becomes zero or the simulation halts. Again we equivalent expect the ISAAC scenario to produce similar behaviour. Parameters were set at $\mathrm{ID}_{u,v} = 10,\ \rho_{u,v} = 0.05,\ \mu_u = (0.2, 0.15)\ \mu_v = (0.3, 0.35),\ \mathrm{IT}_{u,v} = 0.5,\ r_{a,r_{u,v}} = 5,\ D_{u,v} = 5,\ \boldsymbol{C_u} = (60, 60),\ \boldsymbol{C_v} = (-60, -60),\ A_{a_{u,v}} = 5,\ A_{r_{u,v}} = 0.5,\ r_{C_{u,v}} = 5,\ \Delta c = 20,\ ; attack_{u,v} = 1,\ d_{u,v} = 2 \times 10^{-6},\ \beta_{u,v} = 8 \times 10^{-8},\ \nu_{u,v} = 0.2,\ \tau(t = 0) = 10^{-7}$, end time $t = 5 \times 10^{-3}$, $atol = rtol = 10^{-3},\ \Delta x = 0.01$.



(a) Contours of Profiles in 2D

(b) Graphs of profiles 3D

Figure 7.5: Equal Forces, Both Attacking, Forces Offset

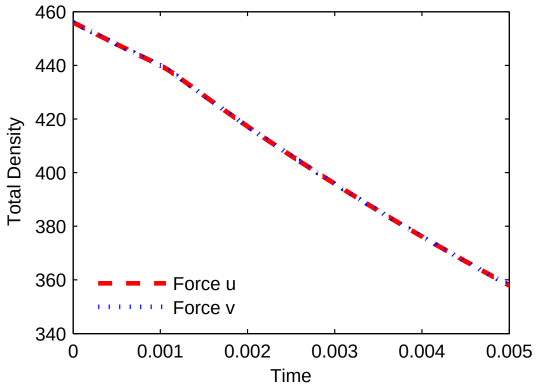


Figure 7.6: Losses for Figure 7.5

No significant change in behaviour is seen between Figures 7.5 and 7.1, however the same avoidance of opposition forces as sub-sections pass each other is seen.

7.2.1 ISAAC Approximation

Parameter	Red	Blue	Parameter	Red	Blue
Squad Size	200	200	Combat	-20	-20
w_1	10	10	Battlefield length	100	
w_2	50	50	Battlefield width	100	
w_3	10	10	Initial Dist Centre x	10	90
w_4	50	50	Initial Dist Centre y	50	50
w_5	0	0	Size x	20	20
w_6	20	20	Size y	20	20
r_S	5	5	Flag x	1	100
r_F	2	2	Flag y	1	100
r_T	2	2	Terrain	no	
w_M	2	2	Move sampling order	random	
Prob Hit	0.002	0.002	Fratricide	no	no
Max Sim tgts	5	5	Reconstitution	no	no
Defence Measure	1	1	Terrain	no	
Cluster	10	10			

Table 7.2: ISAAC Parameters Equal Forces Mutual Attack. All other parameters are set to zero or *no*.

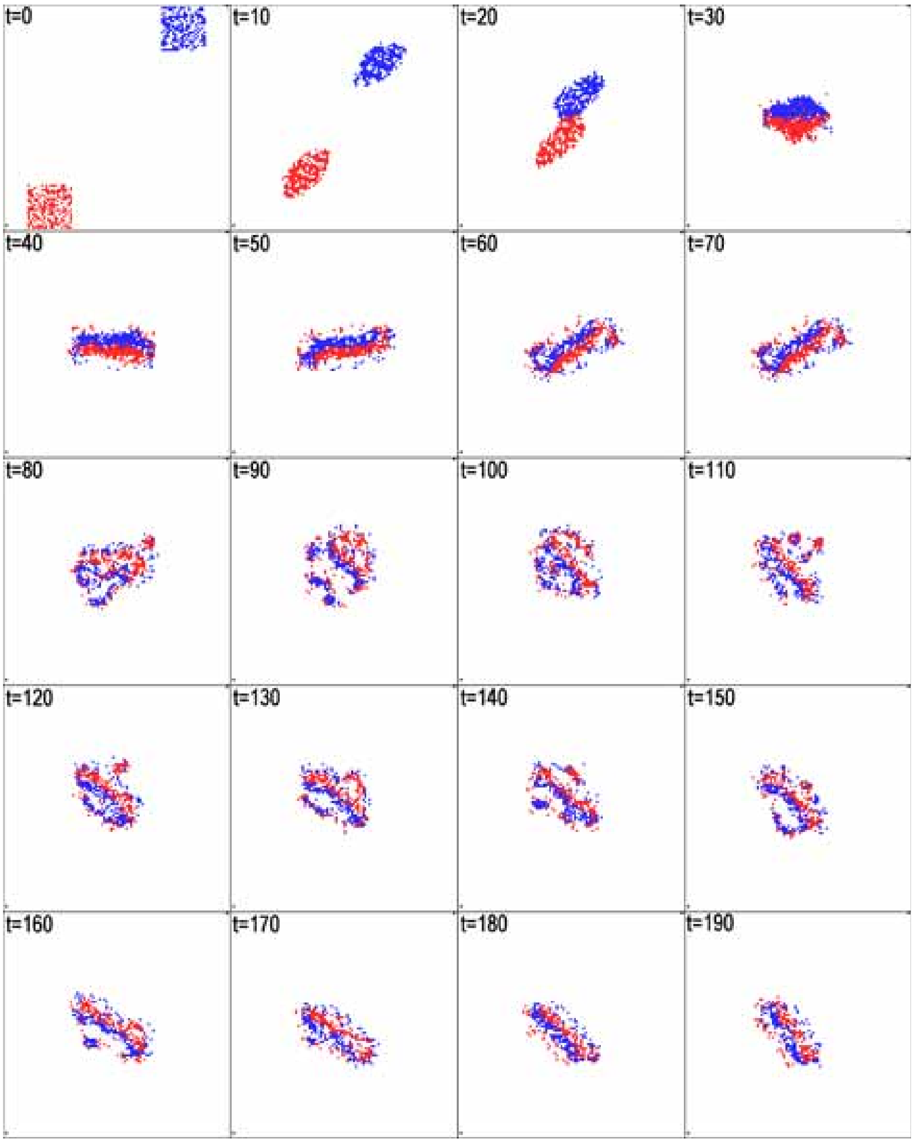


Figure 7.7: Screenshots of ISAAC Approximation to Equal Forces Mutual Attack Scenario

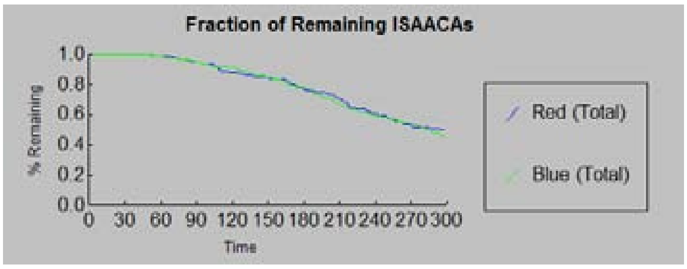
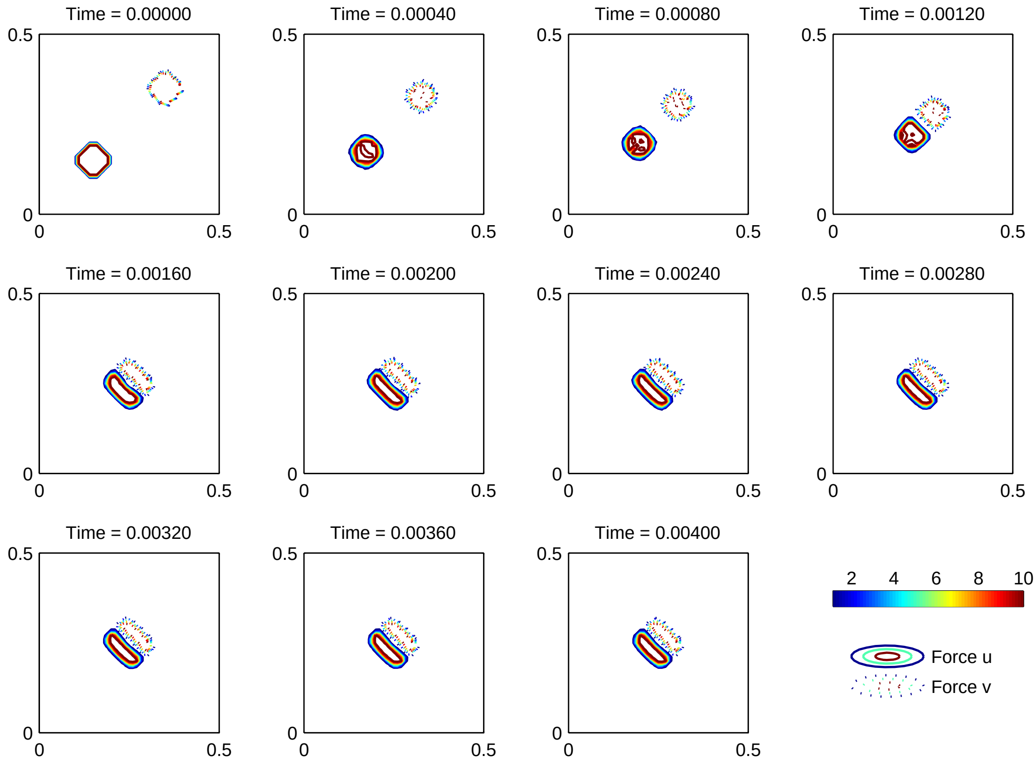


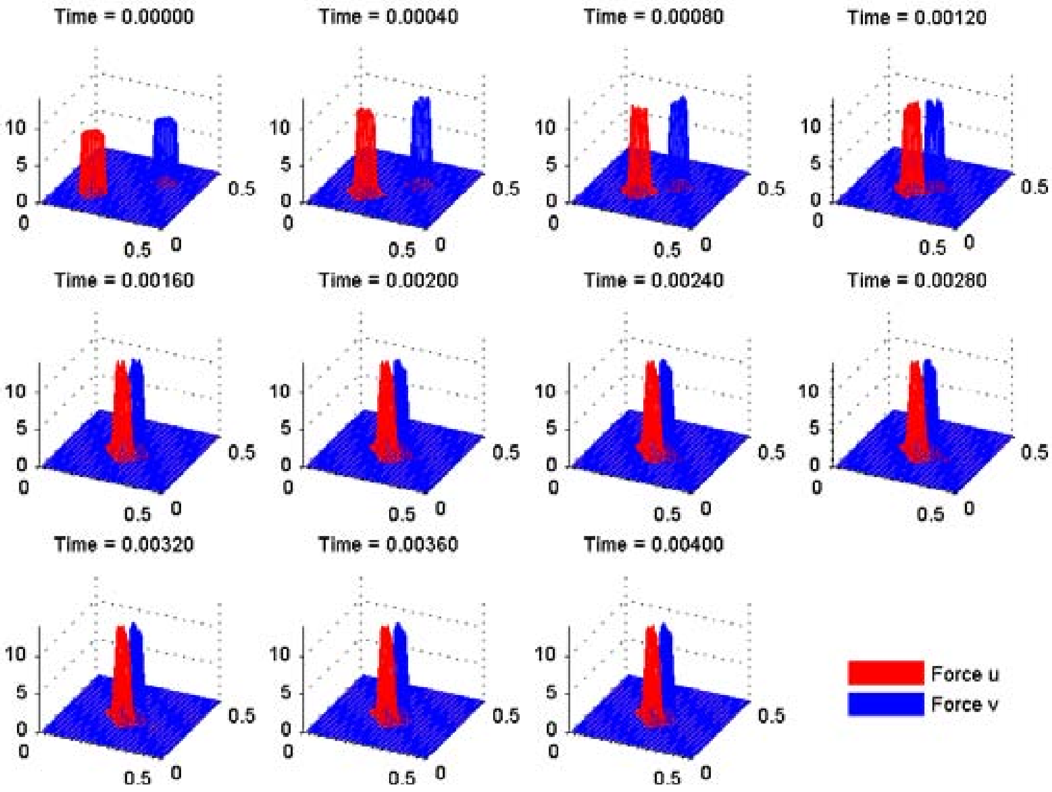
Figure 7.8: Losses for Figure 7.7

7.3 Mutual Retreat

We now investigate the case where both forces have a retreating personality. Changing the value of *attack* to -1 and setting the numerical threshold value Δc to a prohibitively large value should result in some sort of front forming behaviour. As each force should not advance due to the numerical advantage not being reached, this should show similarity to the Classic Fronts scenario in Section (6.4.2). This front forming behaviour is seen as shown in Figure 7.9. Interestingly, by relaxing the numerical advantage threshold to an achievable value, this front forming behaviour is again seen after an initial fragmentation of each force. This fragmentation is due to some areas of the force exceeding Δc and thus the advection term includes an attraction up the gradient of sensed opposite force, while the remaining areas that do not exceed Δc experience a repulsion down this gradient. Reformation of each force occurs after exchanging places. Parameters were set at $\mathrm{ID}_{u,v} = 10,\ \rho_{u,v} = 0.05,\ \mu_u = (0.15, 0.15),\ \mu_v = (0.35, 0.35),\ \mathrm{IT}_{u,v} = 0.5,\ r_{a,r_{u,v}} = 5,\ D_{u,v} = 5,\ \boldsymbol{C_u} = (60, 60),\ \boldsymbol{C_v} = (-60, -60),\ A_{a_{u,v}} = 5,\ A_{r_{u,v}} = 0.5,\ r_{C_{u,v}} = 5,\ \Delta c = 10^{10},\ attack_{u,v} = -1,\ d_{u,v} = 2 \times 10^{-6},\ \beta_{u,v} = 8 \times 10^{-8},\ \nu_{u,v} = 0.2,\ \tau(t = 0) = 10^{-7}$, end time $t = 5 \times 10^{-3}$, $atol = rtol = 10^{-2}$, $\Delta x = 0.01$. For Figure 7.10 Δc was set to 20.

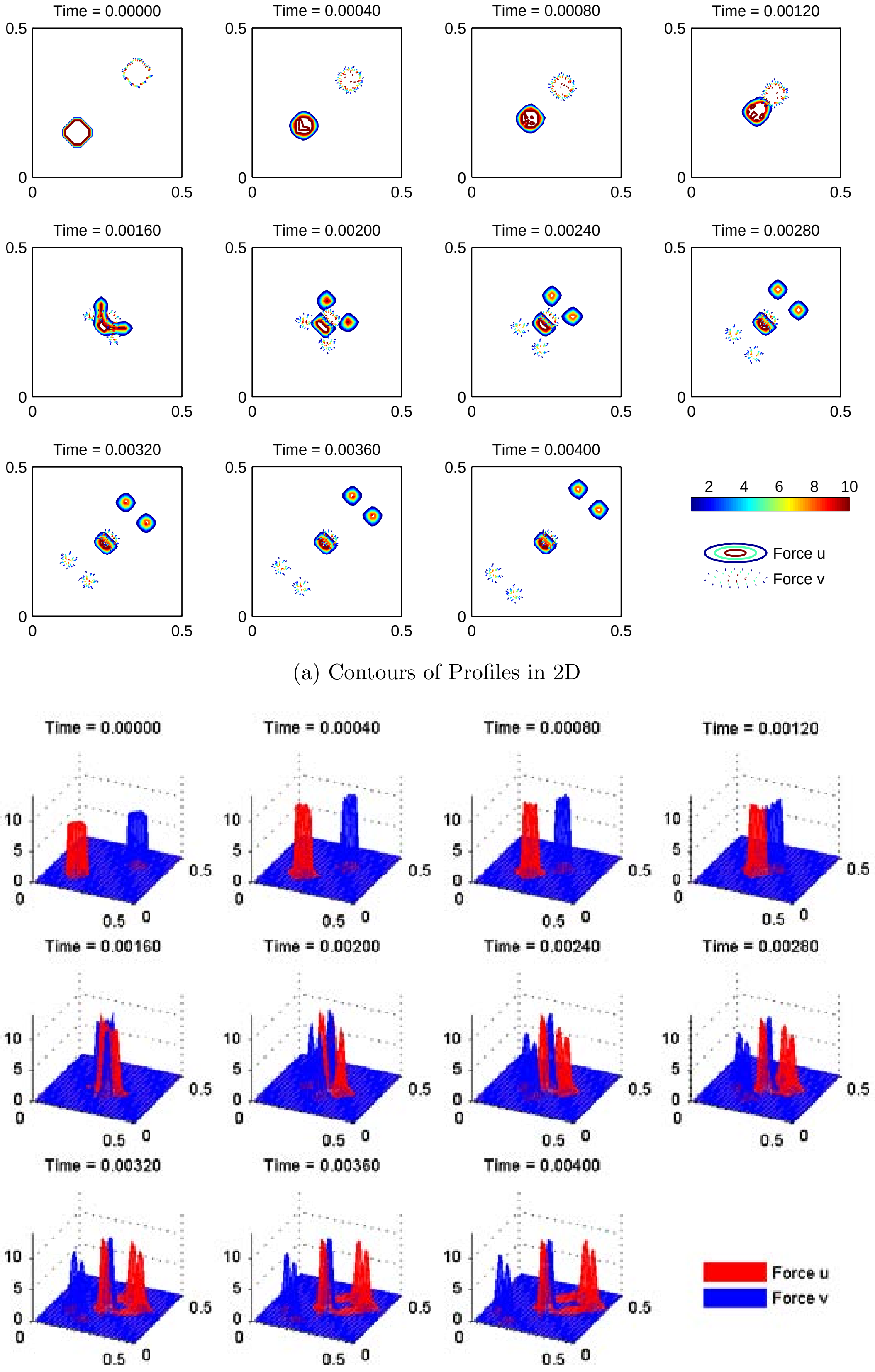


(a) Contours of Profiles in 2D



(b) Graphs of profiles 3D

Figure 7.9: Equal Forces, Both Retreating, High Δc.



(a) Contours of Profiles in 2D

(b) Graphs of profiles 3D

Figure 7.10: Equal Forces, Both Retreating, Low Δc

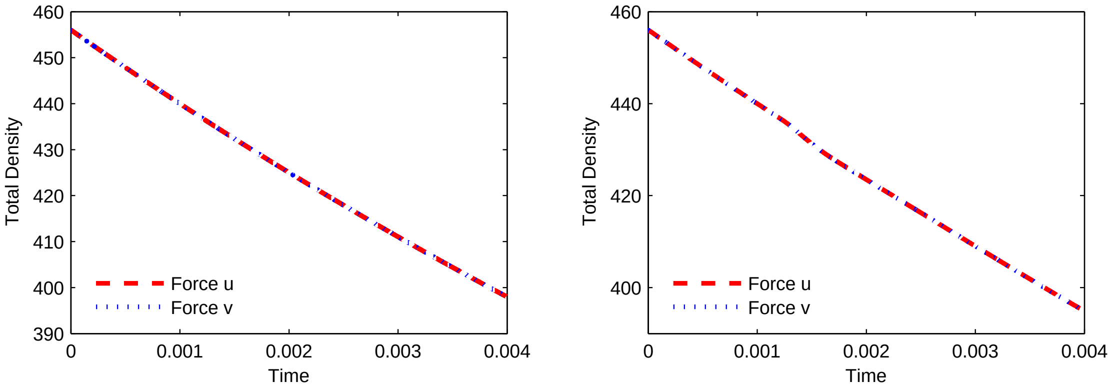


Figure 7.11: Losses for Figure 7.9 and 7.10

Figure 7.10 demonstrates that complicated forms of manoeuvre warfare are possible using a deterministic continuous set of equations with no artificial advection time dependence as used by Protopopescu in order to influence force movement. It is a result of the inherent nonlinearity of the PDEs. This capability is seen as a noted feature of artificial life that is absent from Lanchestrian approaches.

7.3.1 ISAAC Approximation

Again an ISAAC approximation is found.

Parameter	Red	Blue	Parameter	Red	Blue
Squad Size	200	200	Combat	20	20
w_1	10	10	Battlefield length	100	
w_2	50	50	Battlefield width	100	
w_3	10	10	Initial Dist Centre x	10	90
w_4	50	50	Initial Dist Centre y	50	50
w_5	0	0	Size x	20	20
w_6	20	20	Size y	20	20
r_S	5	5	Flag x	1	100
r_F	2	2	Flag y	1	100
r_T	2	2	Terrain	no	
w_M	2	2	Move sampling order	random	
Prob Hit	0.002	0.002	Fratricide	no	no
Max Sim tgts	5	5	Reconstitution	no	no
Defence Measure	1	1	Terrain	no	
Cluster	10	10			

Table 7.3: ISAAC Paramters Equal Forces Mutual Retreat. All other parameters are set to zero or *no*.

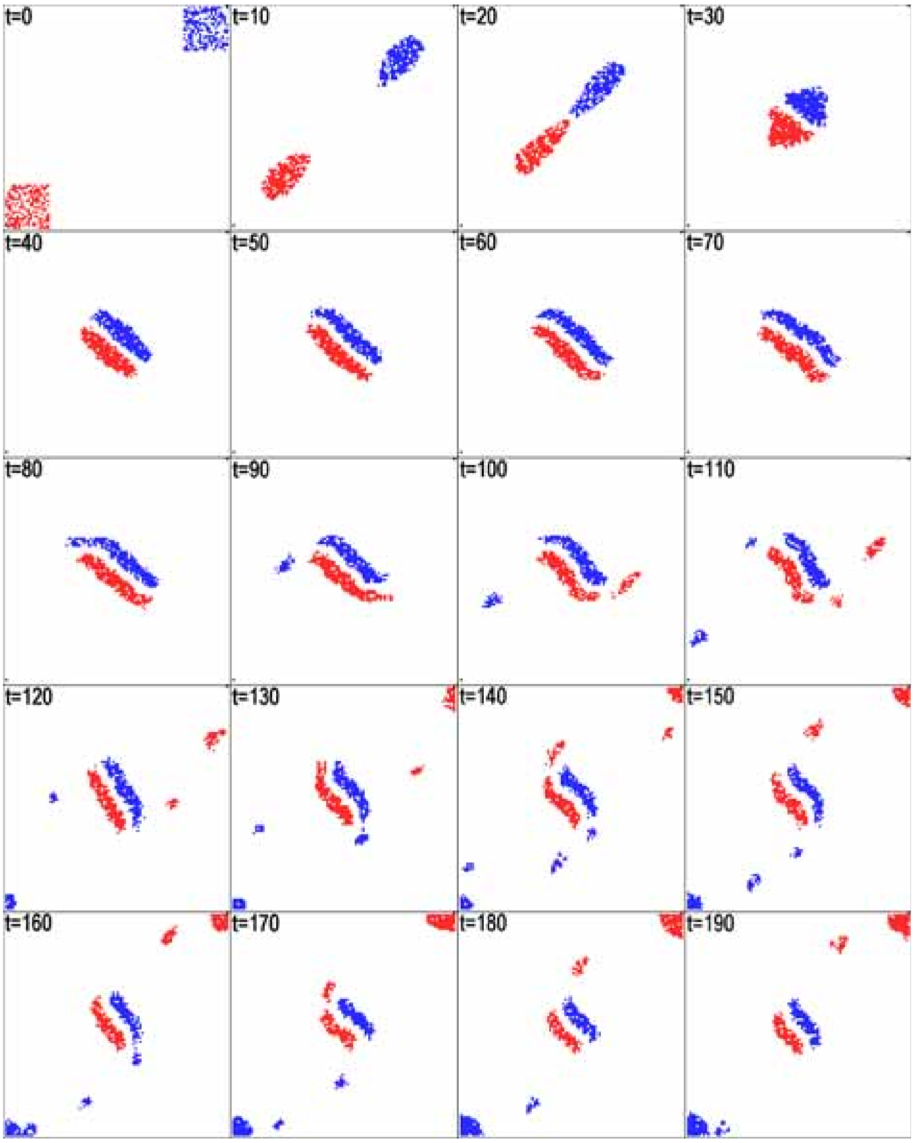


Figure 7.12: Screenshots of ISAAC Approximation to Equal Forces Mutual Retreat Scenario, High Δc.

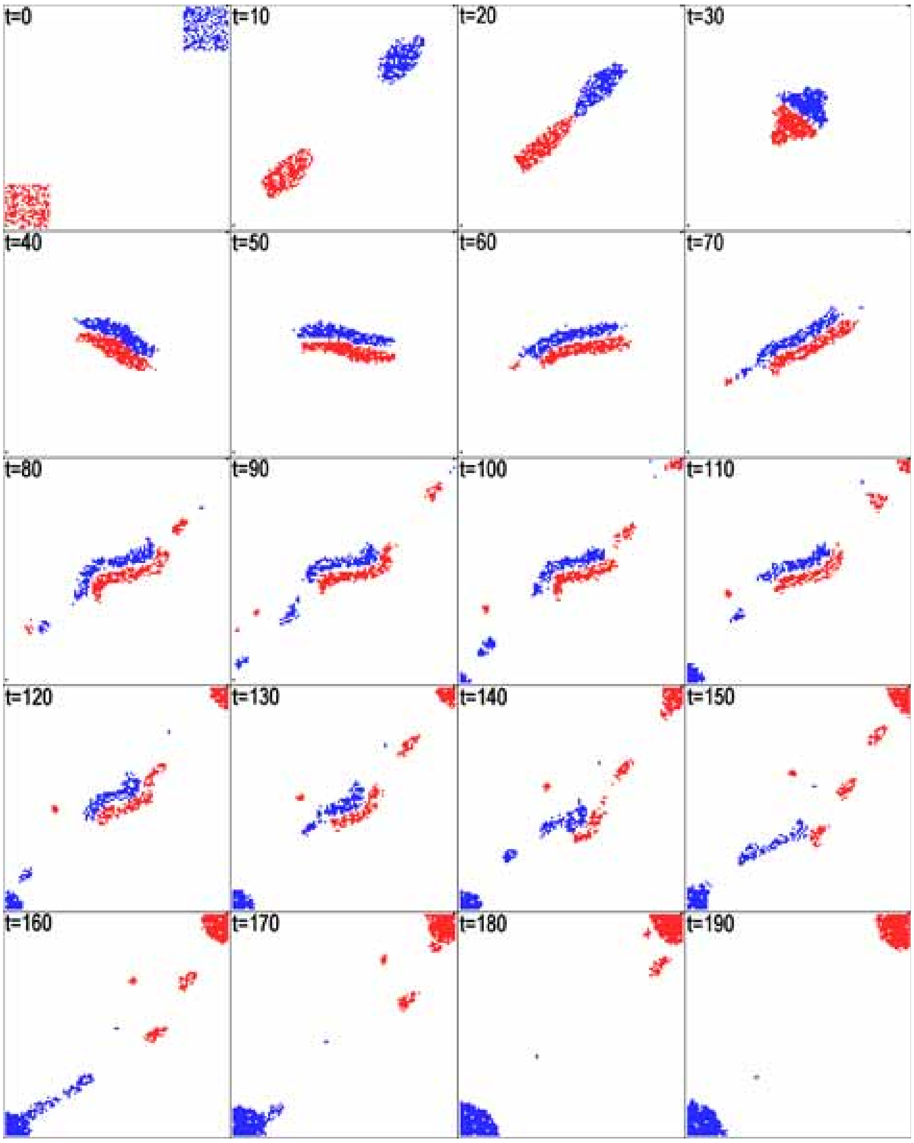


Figure 7.13: Screenshots of ISAAC Approximation to Equal Forces Mutual Retreat Scenario, Low Δc.

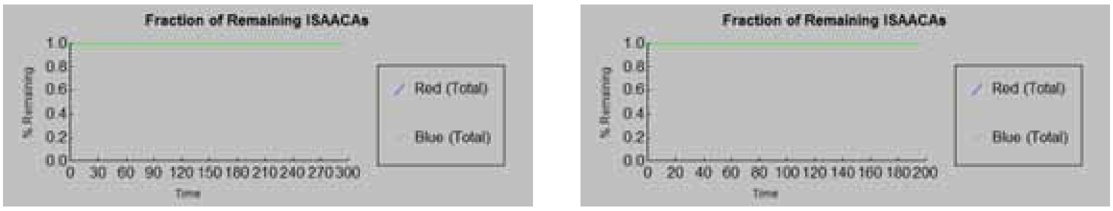
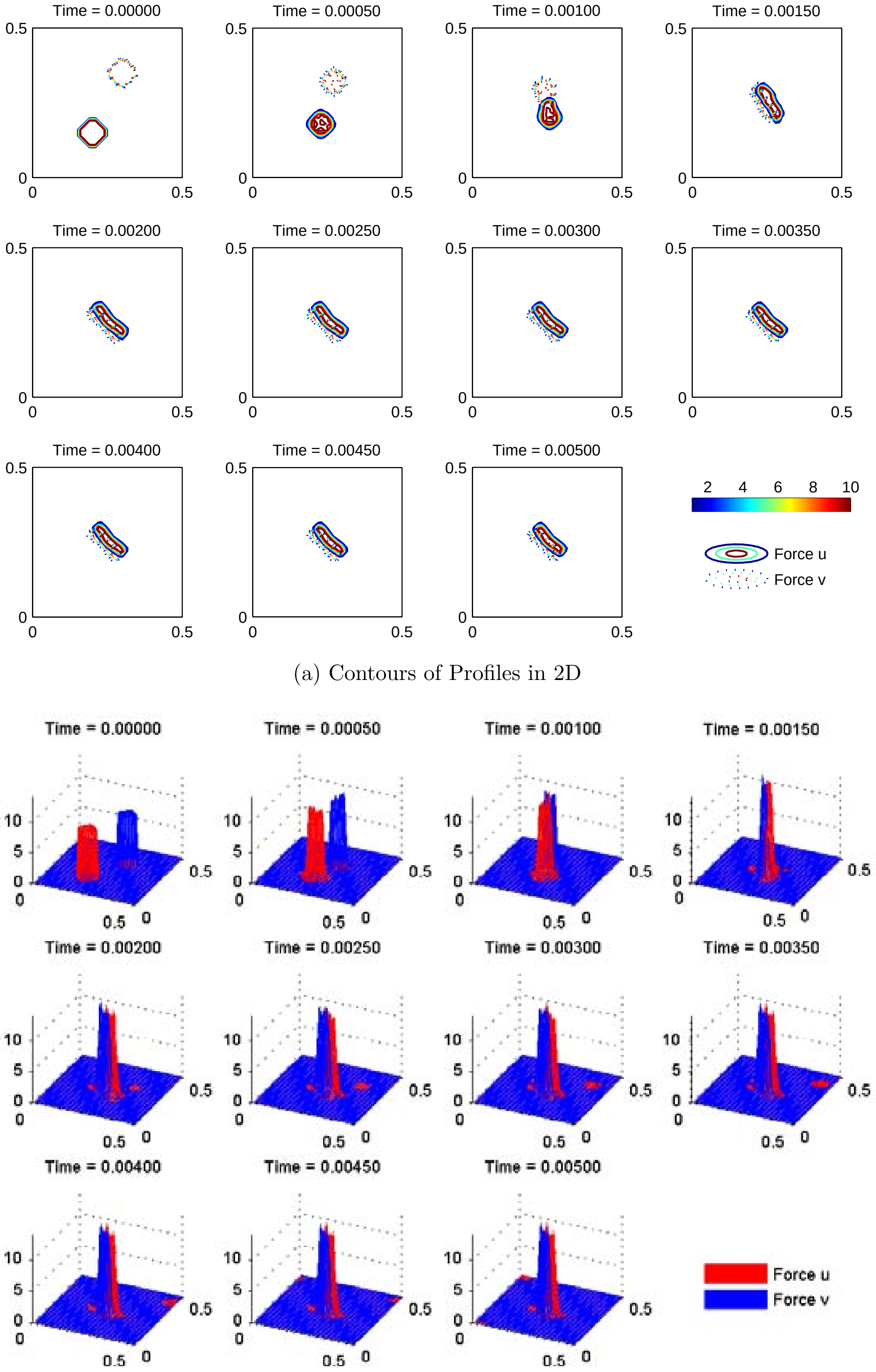


Figure 7.14: Losses for Figure 7.12 and Figure 7.13 respectively.

Note that the front forming behaviour similar to Figure 7.9 is seen in both Figures 7.12 and 7.13. Lowering the Combat value for both ISAAC forces, effectively the lowering of Δc for Figure 7.10 does not result in the fragmentation and reformation of the forces. Only the front forming behaviour is seen. This is our first example of a continuous result that cannot be replicated in ISAAC.

7.4 Mutual Retreat, Offset Initial Position

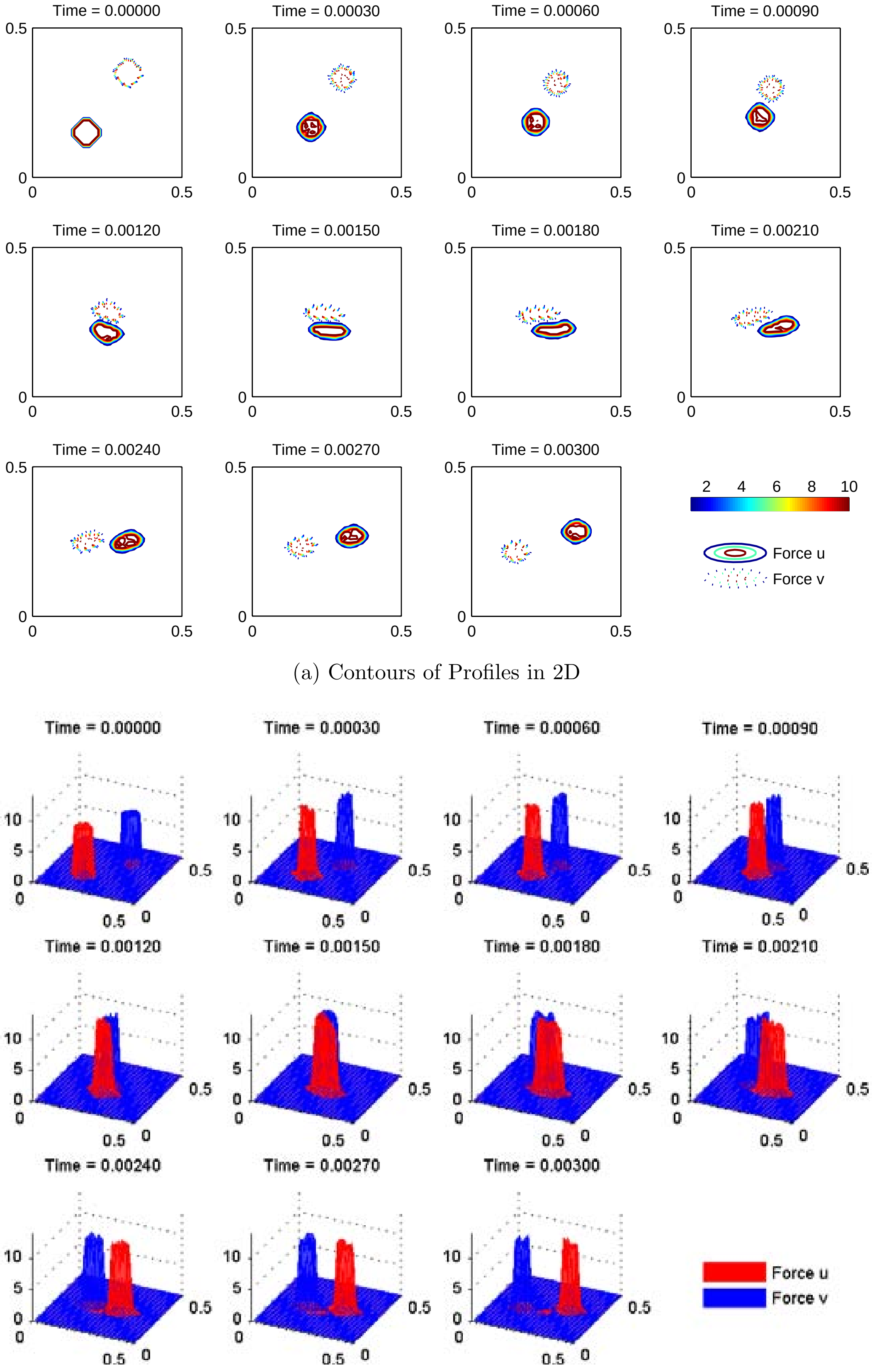
As there are noticeable spatial asymmetries in Figures 7.12 and 7.13, Figures 7.9 and 7.10 are repeated with the initial positions of forces offset. Parameters were set at $\mathrm{ID}_{u,v} = 10,\ \rho_{u,v} = 0.05,\ \mu_u = (0.18, 0.15),\ \mu_v = (0.32, 0.35),\ \mathrm{IT}_{u,v} = 0.5,\ r_{a,r_{u,v}} = 5,\ D_{u,v} = 5,\ \boldsymbol{C_u} = (60, 60),\ \boldsymbol{C_v} = (-60, -60),\ A_{a_{u,v}} = 5,\ A_{r_{u,v}} = 0.5,\ r_{C_{u,v}} = 5,\ \Delta c = 20,\ attack_{u,v} = -1,\ d_{u,v} = 2 \times 10^{-6},\ \beta_{u,v} = 8 \times 10^{-8},\ \nu_{u,v} = 0.2,\ \tau(t = 0) = 10^{-7}$, end time $t = 5 \times 10^{-3},\ atol = rtol = 10^{-3},\ \Delta x = 0.01$. For Figure 7.16 $\Delta c = 10^{10}$ and end time $t = 3 \times 10^{-3}$.



(a) Contours of Profiles in 2D

(b) Graphs of profiles 3D

Figure 7.15: Equal Forces, Both Retreating, Forces Offset, Low Δc



(a) Contours of Profiles in 2D

(b) Graphs of profiles 3D

Figure 7.16: Equal Forces, Both Retreating, Forces Offset, High Δc

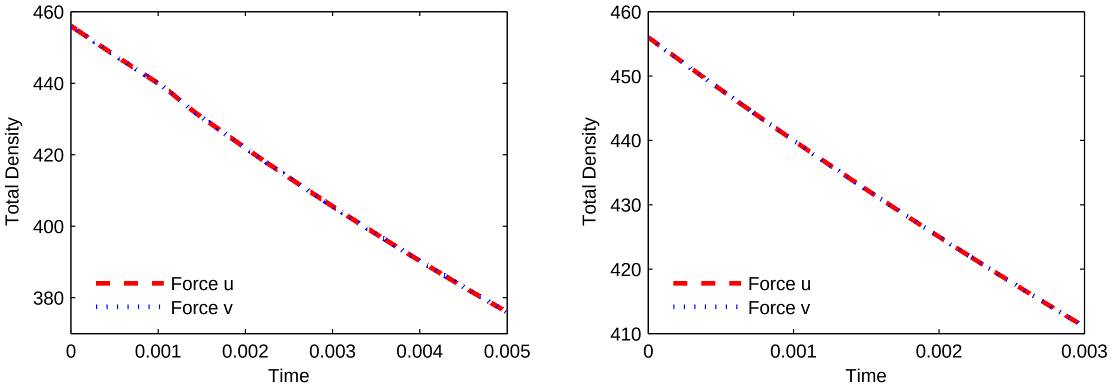


Figure 7.17: Losses for Figures 7.15 and 7.16

Figure 7.15 shows both attractive and repulsive behaviour due to the low threshold value. A stable state is reached due to the dominant attraction at the force periphery keeping the two forces locked in conflict while the dominant repulsion of the force interior prevents the two forces from co-locating. Attrition reaches a constant rate (Figure 7.17b) such that both forces will eventually reduce to zero.

Once the threshold value becomes significant as in Figure 7.16, the weak attractive forces do not dominate the much stronger repulsion resulting in avoidance behaviour for the duration of the simulation. Initial asymmetry determines the direction of force rotation about one another.

7.4.1 ISAAC Approximation

Parameter	Red	Blue	Parameter	Red	Blue
Squad Size	200	200	Combat	20	20
w_1	10	10	Battlefield length	100	
w_2	50	50	Battlefield width	100	
w_3	10	10	Initial Dist Centre x	10	90
w_4	50	50	Initial Dist Centre y	50	50
w_5	0	0	Size x	20	20
w_6	20	20	Size y	20	20
r_S	5	5	Flag x	1	100
r_F	2	2	Flag y	1	100
r_T	2	2	Terrain	no	
w_M	2	2	Move sampling order	random	
Prob Hit	0.002	0.002	Fratricide	no	no
Max Sim tgts	5	5	Reconstitution	no	no
Defence Measure	1	1	Terrain	no	
Cluster	10	10			

Table 7.4: ISAAC Parameters Equal Forces Mutual Retreat, High Δc. All other parameters are set to zero or *no*.

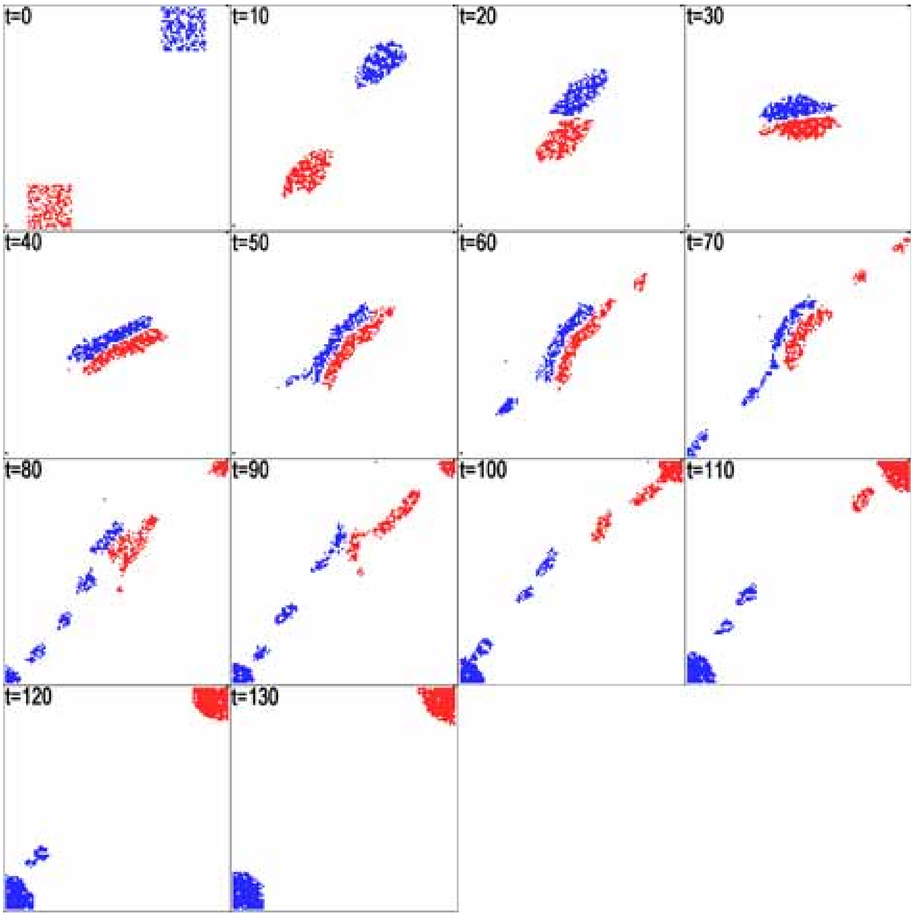


Figure 7.18: Screenshots of ISAAC Approximation to Equal Forces Mutual Retreat Scenario, Low Δc.

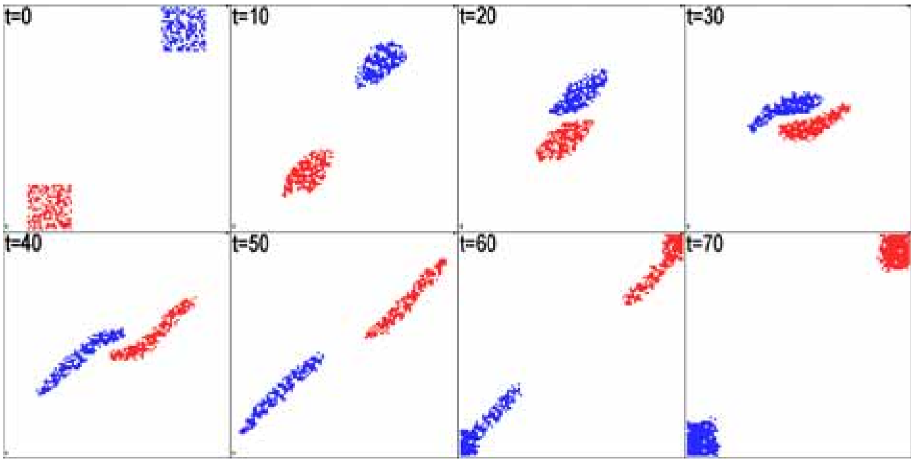
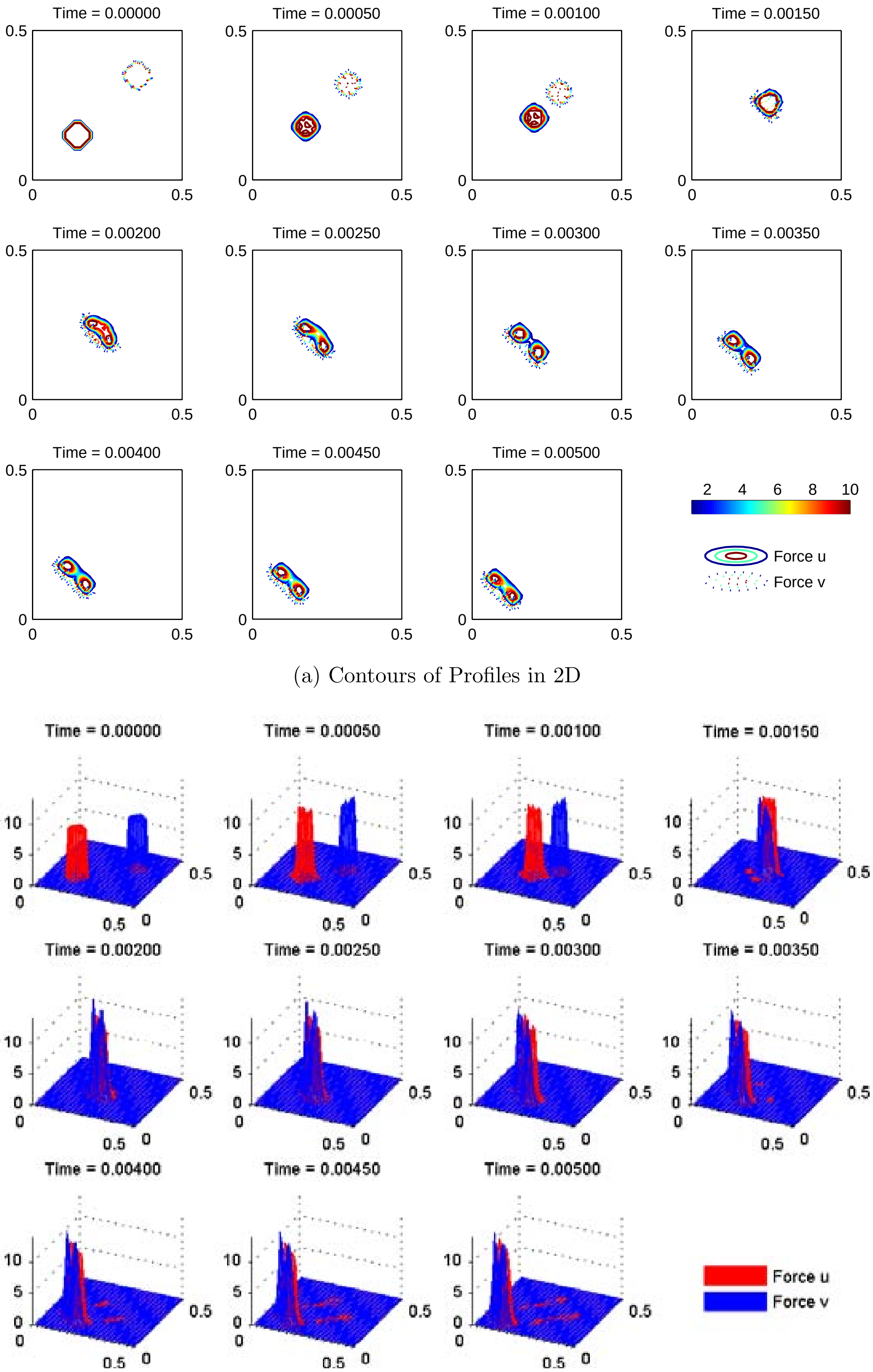


Figure 7.19: Screenshots of ISAAC Approximation to Equal Forces Mutual Retreat Scenario, High Δc, Offset Initial Position.

For this case, finding similar behaviour using ISAAC was easily achieved. Again for a low threshold, attraction towards enemy forces resulted in a semi-stable thin front formation and a constant rate of attrition. However, stochasticity coupled with goal proximity led to the eventual disintegration of this state as the forces moved to their respective goals. For the high threshold value and as expected, strong repulsion prevented any significant interaction with the forces manoeuvring around each other and proceeding to their goals. Note how each force streams towards the goals due to the increasing dominance of the penalty function goal term.

7.5 Attack and Retreat

We now look at the combination of attacking and retreating forces. Setting $\mathrm{ID}_{u,v} = 10$, $\rho_{u,v} = 0.05$, $\mu_u = (0.15, 0.15)$, $\mu_v = (0.35, 0.35)$, $\mathrm{IT}_{u,v} = 0.5$, $attack_{Force\,1} = 1$ and $attack_{Force\,2} = -1$. Parameters were set at $r_{a,r_{u,v}} = 5, D_{u,v} = 5$, $\boldsymbol{C_u} = (60, 60)$, $\boldsymbol{C_v} = (-60, -60)$, $A_{a_{u,v}} = 5$, $A_{r_{u,v}} = 0.5$, $r_{C_{u,v}} = 5$, $\Delta c = 20$, $attack_{u,v} = -1$, $d_{u,v} = 2 \times 10^{-6}$, $\beta_{u,v} = 8 \times 10^{-8}$, $\nu_{u,v} = 0.2$, $\tau(t = 0) = 10^{-7}$, end time $t = 5 \times 10^{-3}$, $atol = rtol = 10^{-3}$, $\Delta x = 0.01$.



(a) Contours of Profiles in 2D

(b) Graphs of profiles 3D

Figure 7.20: Equal Forces, One Attacking, One Retreating

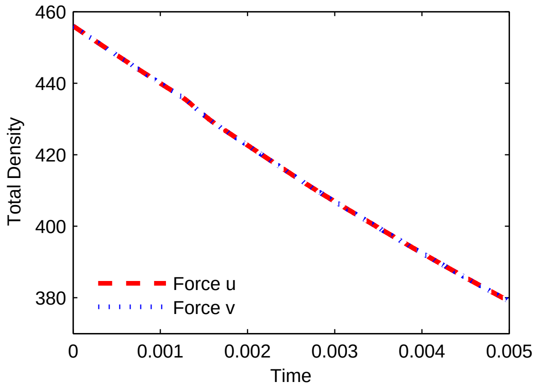


Figure 7.21: Losses for Figure 7.20

Two features are immediately apparent in Figure 7.20 - penetration by the attacking force and its subsequent pursuing of the retreating force. The Blue force (retreating force) behaves in a similar manner as for obstacle avoidance - a bisection followed by a reformation given a sufficient range of attraction and had the Red force been stationary. As the Red force was not stationary, the attack term results in the pursuing of the bisected Blue. As the attack term dominates the attractive/repulsive forces of Red, this too is split into two sections.

Three main periods of constant attrition are seen in Figure 7.21, initial contact $(1.7 - 2e^{-3})$, intact Red $(2 - 3.5e^{-3})$ and split Red $(3.5 - 5e^{-3})$. Despite a visually dynamic simulation, these periods correspond well with Lanchester's original idea of segmenting an overall conflict into a series of smaller ones. Thus each of these three sections can be thought of as a separate Lanchester-type conflict.

7.5.1 ISAAC Approximation

We again seek to find an ISAAC scenario producing similar results.

Parameter	Red	Blue	Parameter	Red	Blue
Squad Size	200	200	Combat	-20	20
w_1	10	10	Battlefield length	100	
w_2	50	50	Battlefield width	100	
w_3	10	10	Initial Dist Centre x	10	90
w_4	50	50	Initial Dist Centre y	50	50
w_5	0	0	Size x	20	20
w_6	20	20	Size y	20	20
r_S	5	5	Flag x	1	100
r_F	2	2	Flag y	1	100
r_T	2	2	Terrain	no	
w_M	2	2	Move sampling order	random	
Prob Hit	0.002	0.002	Fratricide	no	no
Max Sim tgts	5	5	Reconstitution	no	no
Defence Measure	1	1	Terrain	no	
Cluster	NA	NA			

Table 7.5: ISAAC Parameters Equal Forces Both Retreat and Attack. ISAAC Parameters Equal Forces Both Retreat and Attack. All other parameters are set to zero or *no*.

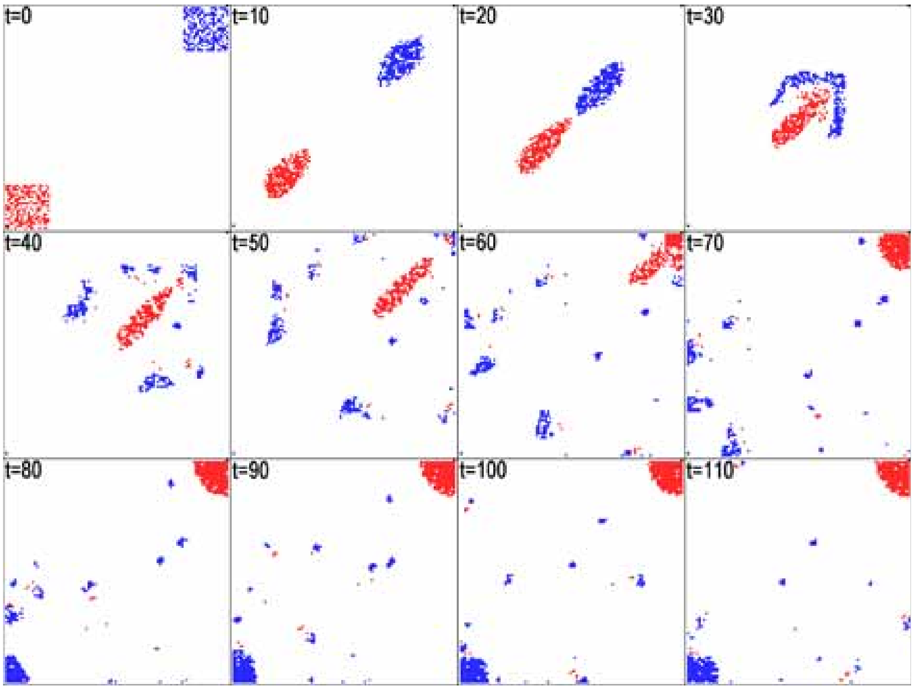


Figure 7.22: Screenshots of ISAAC Approximation to Equal Forces Both Retreat and Attack Scenario

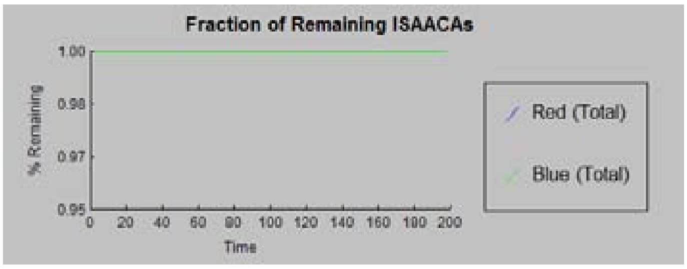


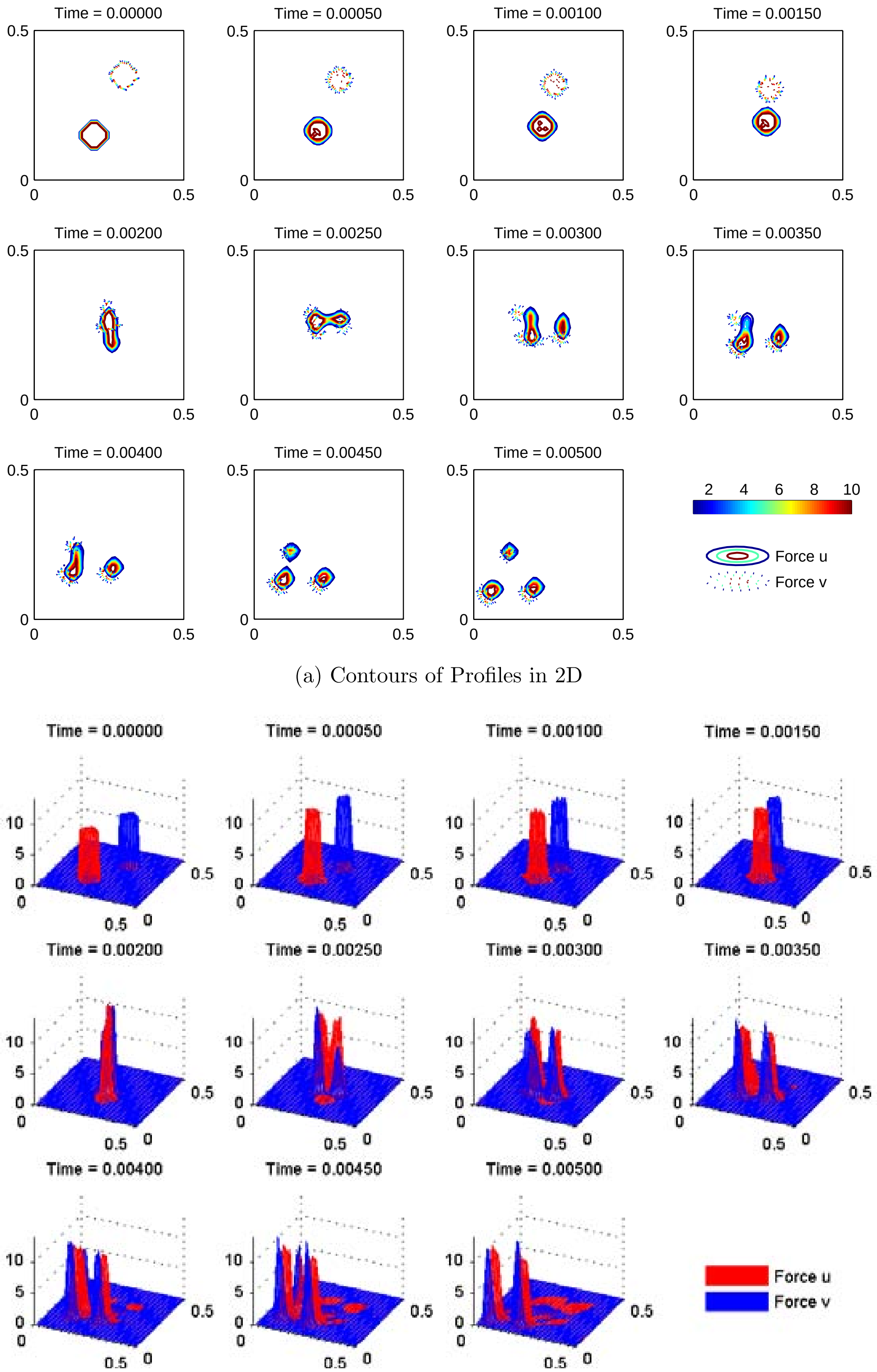
Figure 7.23: Losses for Figure 7.22

Note that the penetration by the Red attacking force is present however the main concentration of the Red force does not continue pursuing. Small elements of the Red force are visible from $t = 220$ onwards pursuing the Blue force. Despite setting for the Red force $w_{2,4}$ to their maximum values of 100 for maximum attraction to Blue, w_6 to a minimum of 1 to ensure the minimal overall velocity in the direction of the Blue goal, and *Combat* to a maximum of -100 to promote maximum pursuing of the Blue force, the main body of the red force proceeds at a constant rate to the Blue goal with only a small number of agents breaking off in pursuit. That is, no significant variation from that seen in Figure 7.22 is seen.

This is another example of the inability of ISAAC to reproduce our continuous results.

7.6 Attack and Retreat, Offset Initial Position

We continue our investigation into the effects of spatial asymmetry through offsetting the initial positions of the forces. $Force\ 1_{attack} = 1$ and $Force\ 2_{attack} = -1$. Parameters were set at $\mathrm{ID}_{u,v} = 10$, $\rho_{u,v} = 0.05$, $\mu_u = (0.15, 0.15)$, $\mu_v = (0.35, 0.35)$, $\mathrm{IT}_{u,v} = 0.5$, $r_{a,r_{u,v}} = 5$, $D_{u,v} = 5$, $\boldsymbol{C_u} = (20, 0)$, $\boldsymbol{C_v} = (-20, 0)$, $A_{a_{u,v}} = 5$, $A_{r_{u,v}} = 0.5$, $r_{C_{u,v}} = 3$, $\Delta c = 100$, $attack_{u,v} = -1$, $d_u = 10^{-5}$, $d_v = 10^{-5}$, $\tau(t = 0) = 10^{-7}$, end time $t = 5 \times 10^{-3}$, $atol = rtol = 10^{-3}$, $\Delta x = 0.01$.



(a) Contours of Profiles in 2D

(b) Graphs of profiles 3D

Figure 7.24: Equal Forces, One Attacking, One Retreating, Forces Offset

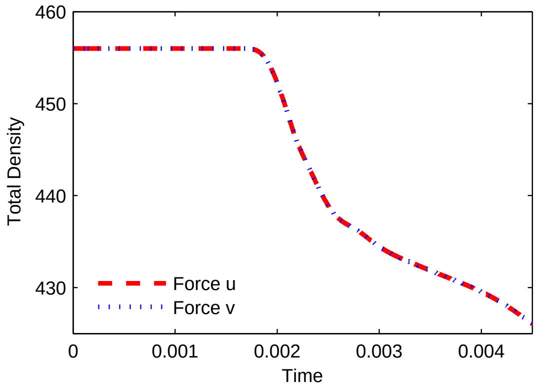


Figure 7.25: Losses for Figure 7.24

This degree of offset of the initial positions results in a precession of the two forces followed by pursuit. At $t = 0.002$ a small section of the Blue force is extended towards the $(0.3, 0.25)$ position. The attack convolution of the trailing section of the Red force then becomes significant in the velocity term resulting in this section separating from the main Red force. For this small Blue section, the retreat gradient combined with the attraction of the main Blue force results in its recombination with the main Blue force.

7.6.1 ISAAC Approximation

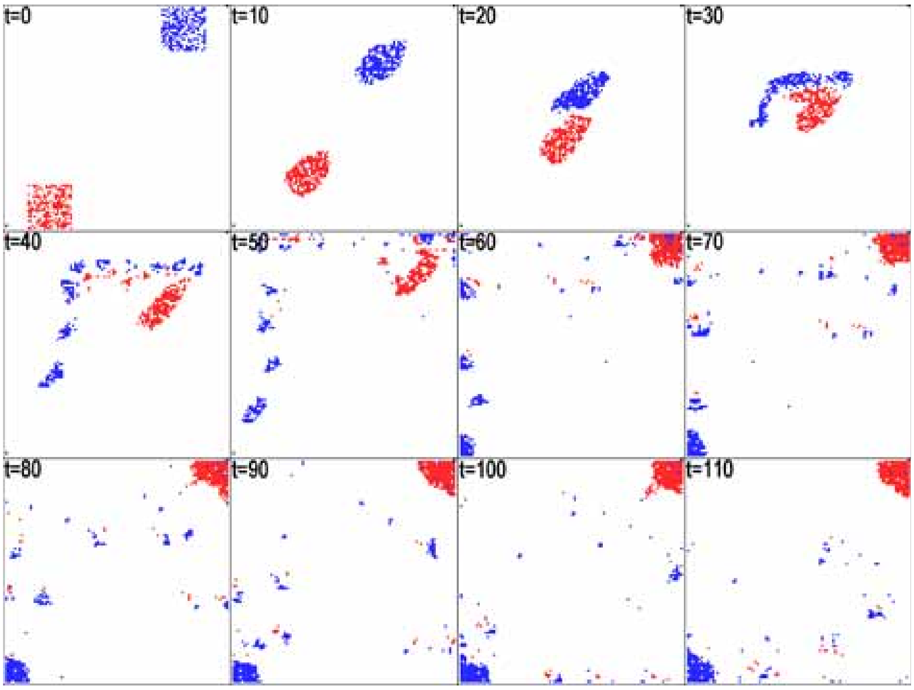


Figure 7.26: Screenshots of ISAAC Approximation to Equal Forces Both Retreat and Attack Scenario, Forces Offset

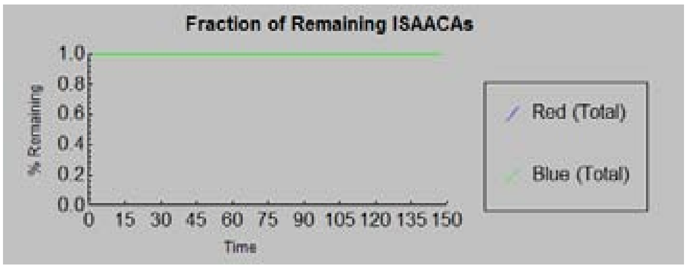


Figure 7.27: Losses for Figure 7.26

Similarly to the previous results, both forces move in an anticlockwise direction around each other due to the initial position offset. Fragments of the Red force then pursue Blue when within sensor range ($t = 80$ in Figure 7.26) however the main body continues to the Blue goal relatively unhindered. There is no redirection of the Red force as a whole as seen in Figure 7.24.

7.7 Equal Forces Discussion

What does this show us? The inability of forces to strongly intermingle or co-locate in ISAAC can result in missing out on observing behaviour as seen in our continuous model that could be classed as manoeuvre warfare. A higher clustering variable may result in a more circular profile, however this is at the expense of the degree of "overlap", "mixing" or co-location possible between forces. Lowering the cluster variable however does produce sparser, less dense profiles allowing greater mixing between forces. The difficulty with using this approach is that an advecting, sparser force tends to form long narrow distributions as seen at $t = 40$ in Figure 7.13, $t = 100$ in Figure 7.19 and $t = 40$ in Figure 7.22, rather than the circular distributions of our continuous model as seen at $t = 0.0008$ in Figure 7.9, $t = 0.0009$ in Figure 7.16 and $t = 0.0015$ in Figure 7.20. Regardless of whether the density of the ISAAC force distributions is high or low, behaviour equivalent to that of the continuous scenarios cannot be consistently replicated from scenario to scenario. This could lead an experimentalist using only ISAAC to conclude that the dynamics for even these simple scenarios are much simpler than our PDE results indicate. The inability to increase the density beyond one soldier per cell must be considered when using a cellular automaton model. One potential improvement for ISAAC could be the ability for each cell to allow multiple agents. This is discussed in more detail in the next chapter.

It could also be argued that the high degree of fragmentation seen in the ISAAC results, for example the Attack/Retreat case, is quite unrealistic. Retaining a high level of cohesiveness would be strongly desired from the viewpoint of self preservation of the force, a tactic also employed by many animal species. A prey species' propensity to fragment is a failing (feature) any predator is likely to exploit. While it is acknowledged that both these modelling approaches are very abstract approximations of combat, a minimal cohesiveness could be seen as a fundamental requirement.

Chapter 8

Density Response Tactic

As mentioned in the swarming literature, the typical individual distance of a particular species is likely be independent on population size yet may be dependent on a variety of other environmental factors. Examples of this include the presence of a predator, time of day or the distribution of food effects may increase or decrease this distance. It follows that the interspacing of soldiers may also have several dependencies - terrain or engagement in combat for example. This potential relationship is not a consideration that is included in ISAAC other than the simplistic Cluster constraint which is constant throughout each simulation. Neither thus far has it been incorporated into our PDE model where only the velocity vector is altered by the presence of an enemy force.

We take inspiration again from the implementation of cellular automata to biological modelling of, for example, fish schooling and predator avoidance of herring [56]. This individual-based model differs from ISAAC in two important ways; (i) a variable density per cell is allowed, and (ii) predator and prey may be co-located in a single cell (predation may only occur when co-located). Repulsion is similar to ISAAC in that it is enforced by an artificial density limit placed on each cell, however this limit was given a predation dependency. If a herring senses a predator within the defined *panic distance*, this maximum allowable density per cell increases. These are two key model constructs that could be employed in both ISAAC and our model to enrich the range of possible behaviours. Firstly, should ISAAC be extended to allow for a density greater than one per cell and dual occupancy of both forces per cell, an increase in mixing or co-location of the two forces may lead to differing scenario results. A revisit of the scenarios presented in Chapter 6 after this modification may result in a closer alignment of behaviours to our model. This feature is already included in our model. Secondly, and this applies to our PDE model also, is the introduction of density dependence on the detection of a threshold number of enemy forces. This dependence should not be to the negation or the detriment of the social interaction dependency as is the case in [56, 57], where the velocity of an individual under attack may be solely determined by the velocity of the approaching attacker with no dependency on conspecifics. Rather, we envisage that attraction/repulsion terms of our PDEs and the Cluster constraint of ISAAC be altered to include this effect. We propose a modification to the attraction/repulsion ratio that determines the interior equilibrium density in order to introduce this concept.

Currently the constant interior density or typical individual distance is determined by the attraction and repulsion values as described in (2.1):

$$F = \frac{A_a}{A_r}. \tag{8.1}$$

There are a variety of ways in which this term could be modified in order to take in to account the presence of enemy soldiers on the target density. A switch as in [56], or a predator influence function in the form of a decay function that smoothly scales the strength of reaction to predators depending upon the separation distance [57]. For IBMs or CAs, repulsion is again artificially imposed through the definition of a maximum density per cell, whether that is one individual per cell as in ISAAC, or several.

For our model we can include this type of dependency through either the attraction, repulsion or both of these terms and we propose the following implementation. Following [56] we define the enemy influence to affect the strength of the attraction parameter A_a once the number of enemy soldiers detected within the sensor distance r_S exceeds the given threshold Δc. Note that this implementation uses the already defined parameters in (7.3) and is a natural extension of this concept. It follows that if the detection of a minimum number of enemy soldiers should alter the velocity of the force, then this same threshold and range are appropriate for defining other enemy based tactical responses.

We define:

$$A_a = \begin{cases} \text{factor} \times A_a & N_{enemy} \geq \Delta c \\ A_a & N_{enemy} < \Delta c \end{cases} \tag{8.2}$$

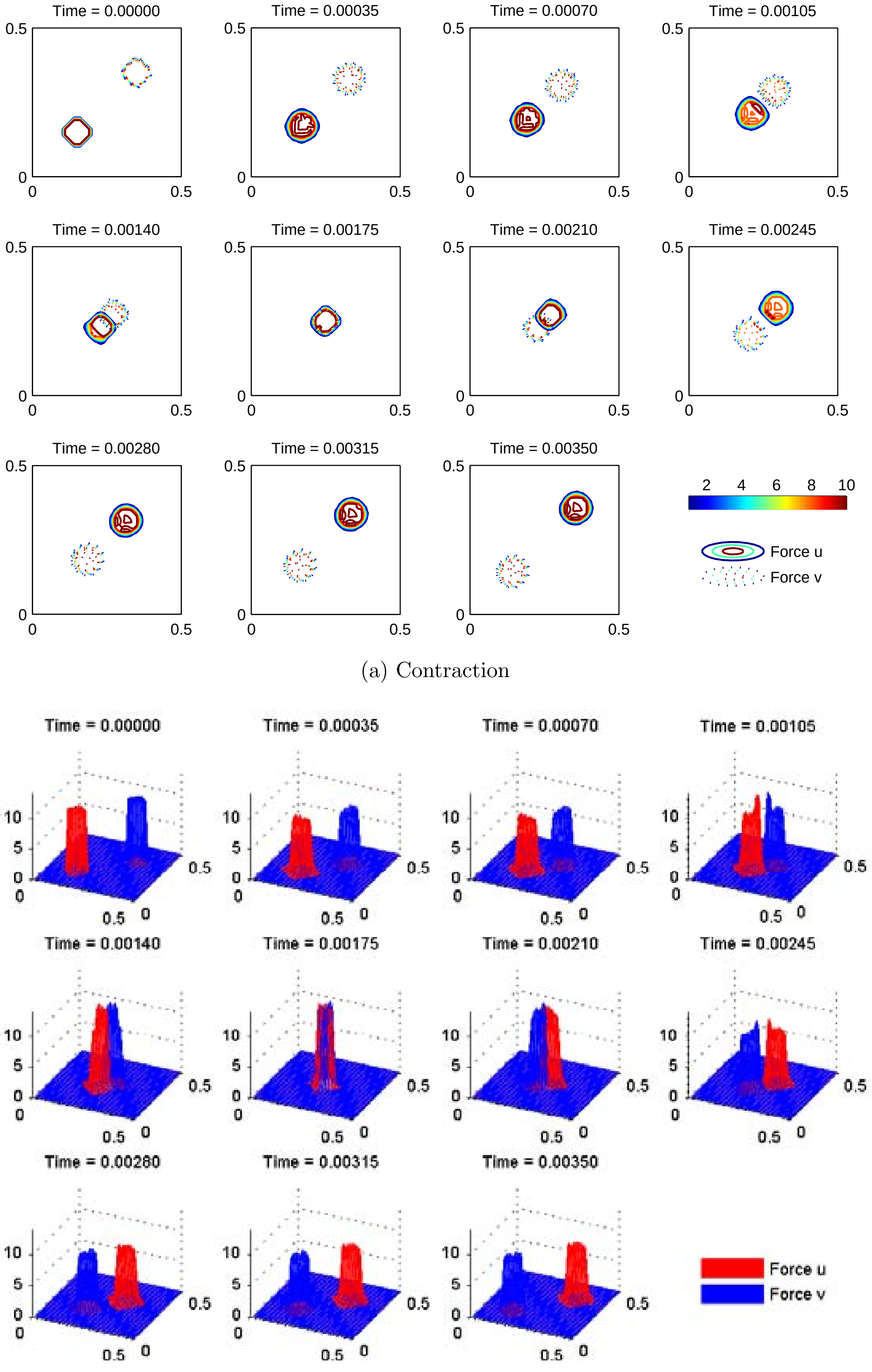
where the factor may be defined as a percentage increase or decrease resulting in a respective increase or decrease in density.

Initially this modification is made and we conduct a series of test cases. Firstly the forces are set to pass directly through one another with an attraction factor for both forces set at 0.8 giving a 20% decrease in the equilibrium density, and then with the factor set at 1.4 or a 40% increase. For clarification, the equations used are:

$$\frac{\partial u}{\partial t} = \nabla\cdot(\boldsymbol{D_u}(u)\nabla u)+\nabla\cdot\{u(\boldsymbol{C_u}u+A_a(K_a{*}u)-A_r u(K_r{*}u))\}+u\,(k_u * v)+d_u v \tag{8.3}$$

$$\frac{\partial v}{\partial t} = \nabla\cdot(\boldsymbol{D_v}(v)\nabla v)+\nabla\cdot\{v(\boldsymbol{C_v}v+A_a(K_a{*}v)-A_r v(K_r{*}v))\}+v\,(k_v * u)+d_v u \tag{8.4}$$

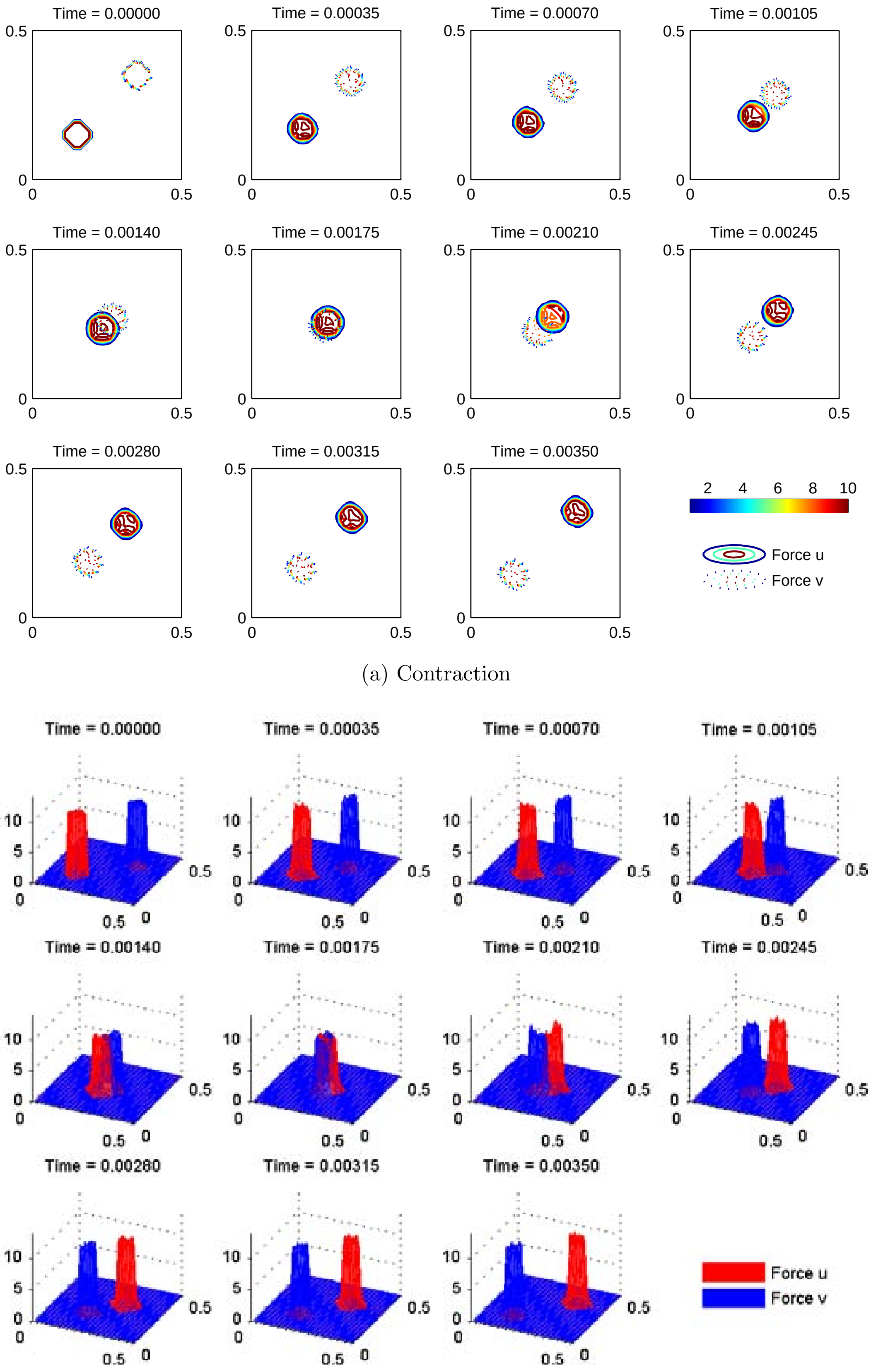
where the attraction A_a is defined as in 8.2.



(a) Contraction

(b) Expansion

Figure 8.1: Effect of attraction factor on force density - Contraction - 2D.



(a) Contraction

(b) Expansion

Figure 8.2: Effect of attraction factor on force density - Expansion - 3D.

Figures 8.1 and 8.2 show quite clearly the changes in density associated with the attraction factor. In the accompanying videos, the propagation of the density change can be seen initiating at the front of the forces, smoothly moving to the rear as the two forces advance closer to each other. This propagation reflects the limits of the sensor range threshold level.

We now consider a simple scenario in order to investigate the relationship between this type of alteration to tactics and density loss through the inclusion of aimed fire. Both the expansion and contraction versions are compared to a baseline scenario.

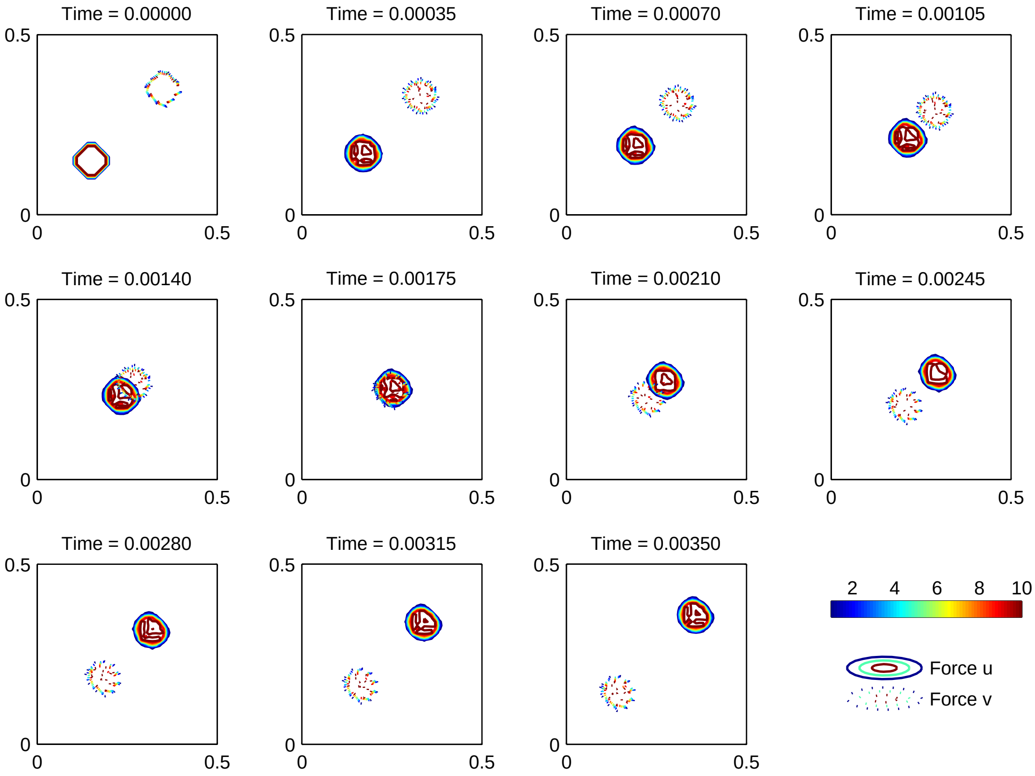


Figure 8.3: Effect of attraction factor on force density - Baseline.

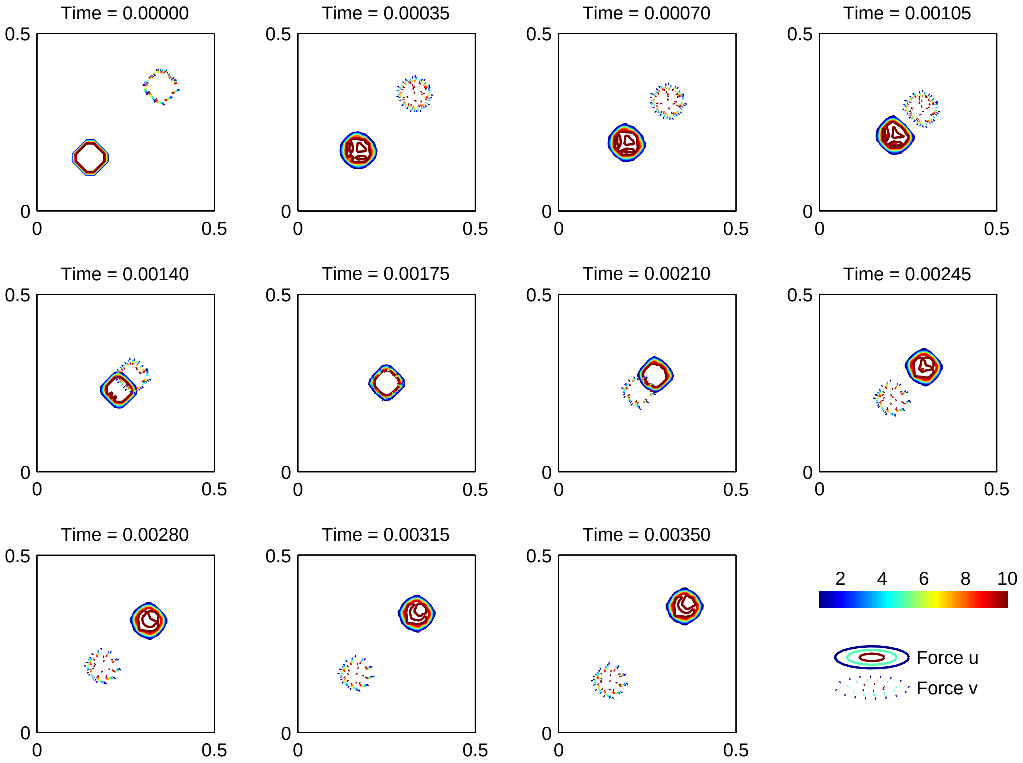


Figure 8.4: Effect of attraction factor on force density - Contraction.

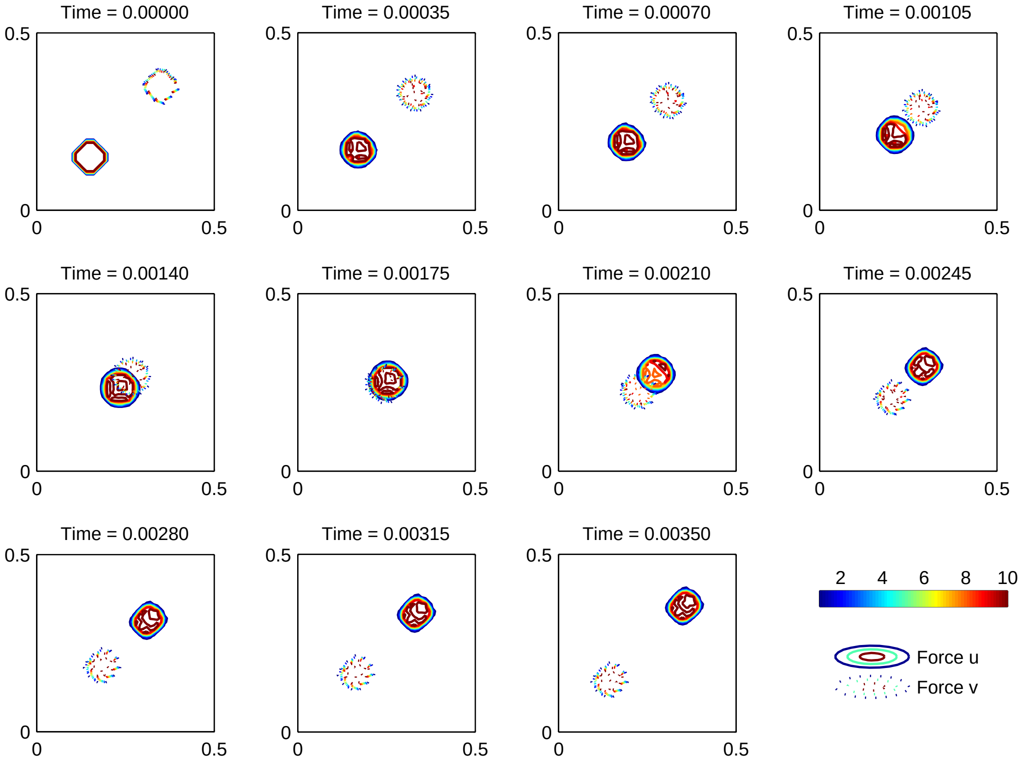


Figure 8.5: Effect of attraction factor on force density - Expansion.

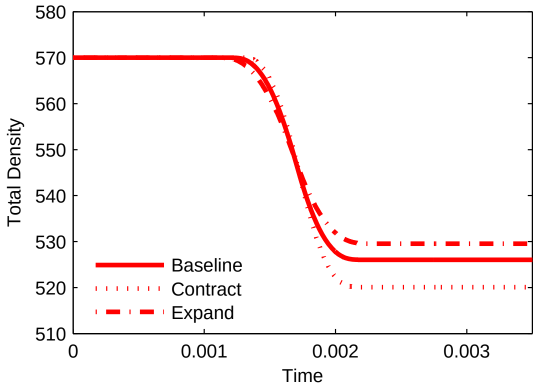


Figure 8.6: Losses for Figures 8.3, 8.4 and 8.5

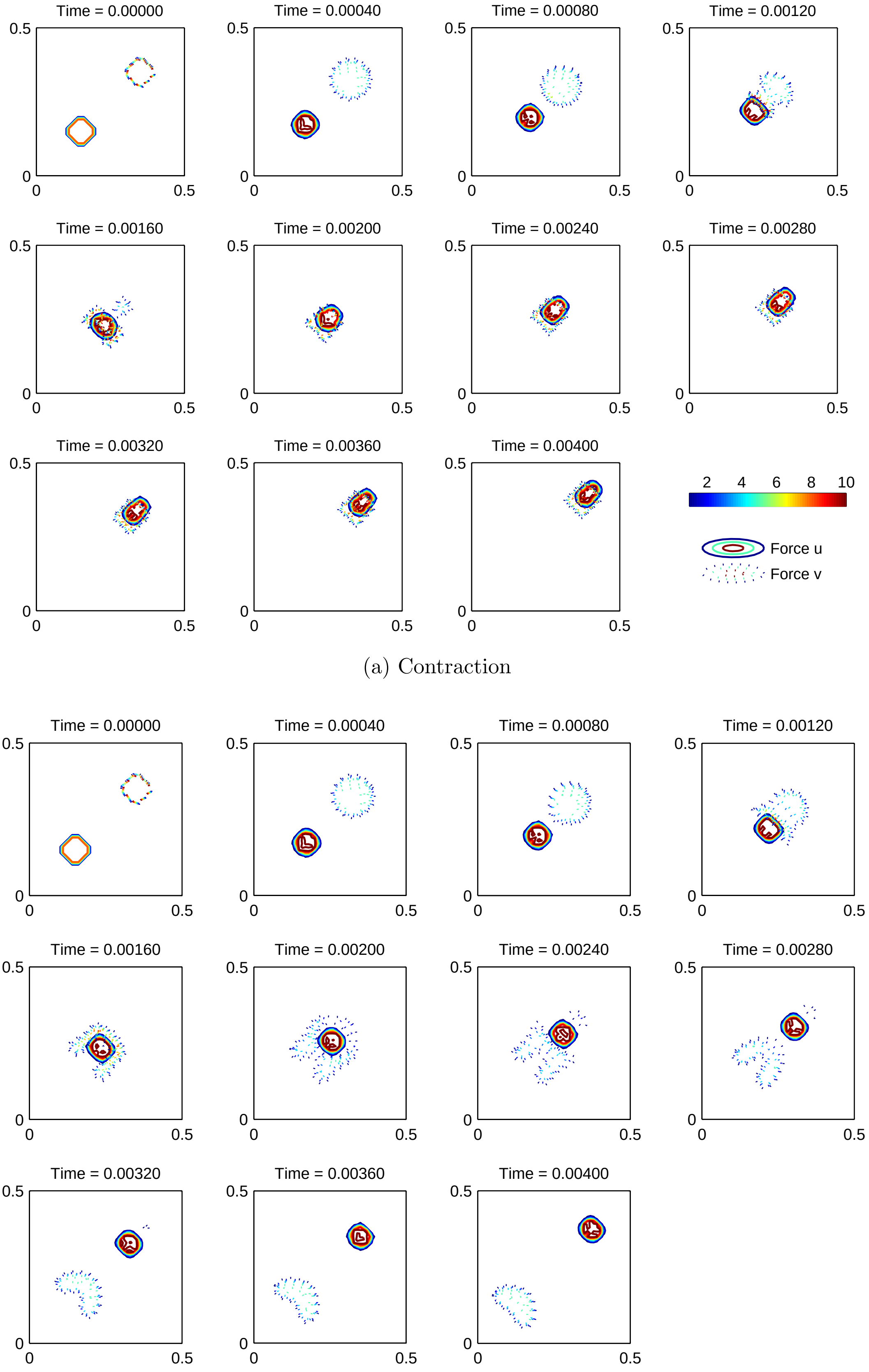
We omit the losses for the second force for clarity as both forces are equal in parameters and suffer the same reduction in density. Although using the expansion tactic results in greater initial density loss, this approach leads to the lowest overall reduction. This earlier commencement of attrition is due to the expansion of the front section of the force that is first to sense an enemy density exceeding the given threshold. Such an expansion then results in a more rapid co-location of both forces and hastens the onset of attrition by aimed fire. Attrition continues for a longer duration as these expanded force distributions remain in contact for a longer period of time. Despite this longer duration, the lower density produces lower attrition rates and the lowest overall force reduction.

Using a contraction tactic produces the opposite behaviour to the expansion tactic. Similarly to the expansion tactic and as expected, contraction propagates from the front to the rear of the force, delaying the onset of force co-location and thus attrition. A higher attrition rate is experienced due to the higher density, and despite the shortened duration of contact, gives the greatest overall density reduction.

Here we have shown that a simple scenario can demonstrate a different possible outcome for each permutation of the density response tactic. Although these outcomes are relatively straightforward in this example, the inclusion of this tactic in our already complex PDE model of (2.6), (2.7) and (6.6) would warrant detailed numerical approximations and further study to ascertain the extent of the effects on the dynamics of the model. As this is beyond the scope of this research, we present here one example from our comparison to the ISAAC scenarios.

8.0.1 Implementation of Density Response Tactic to ISAAC Comparison Scenario

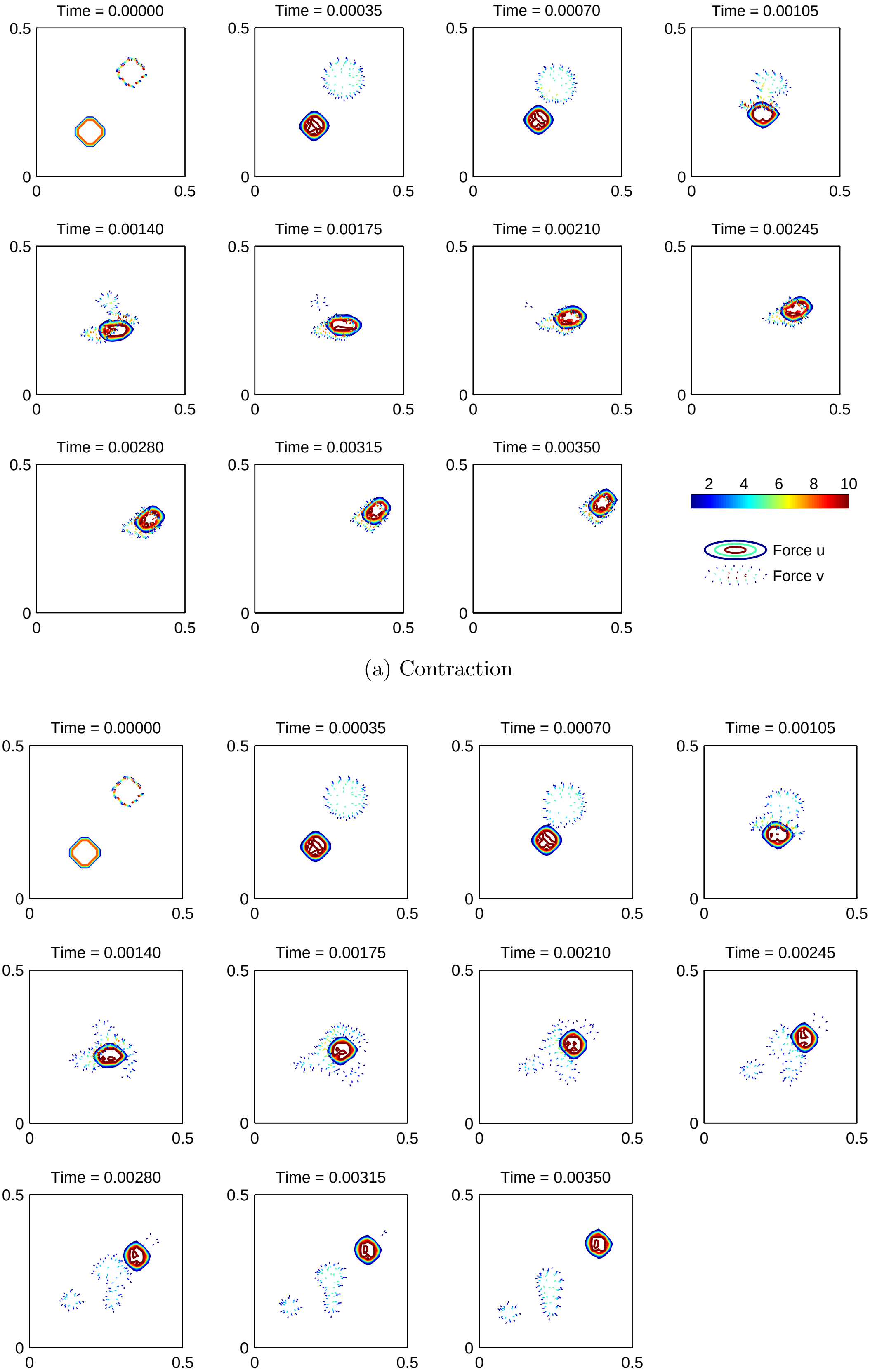
We now revisit one of the comparisons made to the ISAAC scenarios using our PDE model and introduce the density response tactic to the Precess scenario as an example of how this changes the observed dynamics. Firstly each force will undergo expansion upon sensing enemy density above the given threshold, then contraction. Recall that the Precession scenario consisted of both typical non-precession behaviour and atypical precession behaviour when the force density profiles were sufficiently offset. We implement both forms of the density response tactic to both behaviour types.



(a) Contraction

(b) Expansion

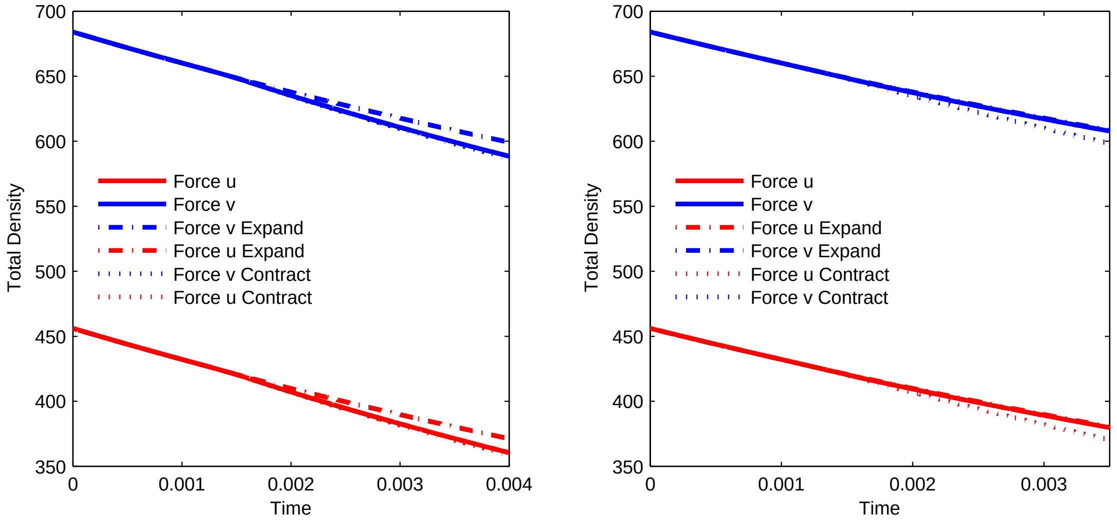
Figure 8.7: Effect of attraction factor on ISAAC Precess Scenario Approximation shown in Figure 6.11.



(a) Contraction

(b) Expansion

Figure 8.8: Effect of attraction factor on ISAAC Anticlockwise Precess Scenario Approximation shown in Figure 6.11.



(a) Precession

(b) Anticlockwise Precession

Figure 8.9: Losses of original ISAAC Precess Scenarios compared with additional attraction tactic.

For the expansion tactic, the lower density and thus larger area of the Blue force causes greater percentage of this force to lie beyond the sensor range used for the attacking term, giving a zero contribution to the velocity vector. So this portion of the force continues on to the opposite corner. Social forces dominate the attacking section of the Blue force and the Red force is not pursued. This is seen in both the typical (Figure 8.7b) and atypical results (Figure 8.8b). As expected there is less density loss for expansion due to the reduced co-location and lack of pursuance (Figure 8.9a) in the Precession comparison, though this difference is negligible for the Precession Offset comparison (Figure 8.9b).

For the contraction tactic however, the higher density induces a smaller overall density profile of the Blue force. This ensures that the Red force then remains entirely within the Blue's sensor range so that the attack term then dominates the Blue force $\mathbf{f}_{\mathbf{vel}}$ term. Again this behaviour is seen in both results (Figures 8.7a and 8.8a). As expected the contraction tactic gives comparable or slightly greater density loss than the baseline.

We can continue with further combinations of expansion and contraction tactics, for example, the Red force adopting expansion and the Blue force contraction, or vice versa. However we leave this and the extension of the density tactic to the remaining ISAAC scenarios for exploration in future research.

This chapter demonstrates the inherent flexibility of our model to incorporate variations to the spatial dynamics or social interaction terms that form the main influence on the observed dynamics. The spatial dynamics terms can be though of as the tactics employed by a force, and forms one of the fundamental drivers in military research - what tactics increase the likelihood of achieving the desired outcome? This equally applies to both forces as an assumption of a completely predictable and static enemy is fraught with danger. By allowing flexibility in the repertoire of tactics that can be explored, a better understanding of combat dynamics may be gained prior to the introduction of additional features such as command structures (as is possible in ISAAC).

Chapter 9

Discussion, Summary and Future Work

9.1 Discussion

Cellular automata can be difficult to use for understanding the underlying dynamics of combat as stochasticity can hinder the extraction of conclusions from a model. All scenarios presented here highlight the dangers associated with attributing *intelligent* reasoning to behaviour shown, when this can be explained quite simply through the effects of the terms in our equations (2.6) and (2.7) and the spatial distribution of forces. This can be seen quite simply in the Classic Fronts scenario. Ilachinski describes the occurrence of the forces passing by one another as the agents having *"found" a way to sneak around.* We believe it is the differences in distribution caused the inherent randomness of ISAAC that results in agents located at the end of the distributions becoming sufficiently distanced so that repulsive forces from enemy agents greatly reduces with respect to the attractive forces of the goal. This dominant attractive force then results in those agents progressing towards the goal. Although the attraction/repulsion of goals in our equations is not weighted with respect to separation distance as it is in the ISAAC penalty function, a slight offset or asymmetry of initial distributions demonstrates this same type of behaviour.

This can also be seen clearly in the Precess scenario where the infrequently observed precession behaviour arises from asymmetries of the force profiles upon commencing combat, rather than being an unexplainable emergent behaviour. The majority of ISAAC Precession simulations show behaviour similar to those obtained from the continuous equivalent without the initial distribution offset as shown in Figure 6.8. Again the observed asymmetries are due to the stochasticity of ISAAC present in the movement and attrition algorithms, and are responsible for the precession seen in Figure 6.10 and [18]. When mimicked in our continuous equations through initially offset distributions, precession was observed. In this case the underlying behaviour without the effects of stochasticity highlighted the similarity between the Precession and Circle scenarios. Again Ilachinski infers a degree of intelligence or emergence by stating that there are *a few stray 'leakers' and an occasional group of a few Blue ISAACAs that choose[s] to leave the main battle and head toward Red's flag.* This is similar to the comments made for the Classic Fronts scenario and our explanation for that behaviour - differences in distribution result in agents becoming sufficiently distanced from enemy agents such that the goal terms in the penalty function become significant.

Our deterministic approach encapsulates basic motivational factors and demonstrates a variety of spatial behaviours. Cohesive troop movement has only been achieved artificially in previous work: by using a desired initial distribution, low diffusion constant, and a sufficiently short overall simulation time, excessive diffusion is prevented and the troop profile cannot diffuse to a unrealistic spread. We have demonstrated that by using relatively simple and physically meaningful form of partial differential equations, cohesive troop movement can be achieved and maintained, even when suffering loss of density through fire. Forces and firing coefficients remain homogeneous, a criticism of the traditional Lanchester approach, yet the nonlinear nature of the equations are able to mimic those seen in ISAAC that have been labelled as complex. A continuum of forces is able to behave in a manner similar to a collection of individual autonomous agents, and shows decentralised self-organisation and adaptation of tactics to suit a variety of combat situations. This is a significant step toward developing a set of realistic continuous equations for combat modelling.

Lauren [24] states that complex adaptive models of combat, such as ISAAC or MANA, display a rich variety of behaviour, a battlefield that is no longer linear and agent adaptivity. How exactly do the agents perform this adapting? The evolution of each agent's position is determined by a penalty function that remains unchanged throughout the entire simulation. There is a danger in the anthropomorphisation of agents, insinuating agents have reasoning and planning abilities when this is obviously not the case. Also these types of wargames concentrate heavily on the addition of extra communication ability between agents, shifting the emphasis to global or increasingly complex nonlocal features. MANA includes many more states and subsequently many more triggers need to be defined to facilitate switching between these states. Increasing the number of required parameters can cloud the process of deriving insight, a danger which has been shown in many of the basic scenarios presented here. For example the removal of parameters such as the *Advance* constraint in the Precess scenario did not prevent the reproduction of similar precession behaviour. Lauren also states that *conventional combat models behave largely as a series of attrition-driven fights* and *ISAAC entities will only fight if conditions are suitable*. If our model is viewed as a conventional combat model as it is based on using Lanchester firing terms, it can be argued that it also behaves as an ISAAC model due to the form of the Spatial Dynamics terms (7.3). Conversely it could be argued that ISAAC is a type of conventional combat model as the movement of each agent is determined in a comparable way to our continuous model with the inclusion of randomness.

Considering the popularity of cellular automata based wargames in military and complex adaptive systems research, it is imperative that complementary avenues of research are undertaken in order to gain a greater insight into the nature of combat. For a research topic such as combat modelling, emphasis of rare events or a misunderstanding of observed behaviour can have significant consequences. Our numerical analysis of this continuous set of equations can provide an alternative explanation to the seemingly intelligent behaviour demonstrated in these agent based scenarios as the same essential nonlocal interactions are present. The need for many multiple agent simulations and the application of data mining techniques becomes much reduced with the simultaneous use of a continuous model. MANA, much like ISAAC, can require approximately 600 runs to establish a mean result for some scenarios [26]. Using the approach employed here it is much easier to establish a mean behaviour with our continuous form.

By treating weapons systems in the simple forms as presented here, behaviour is less likely to be obscured by potentially highly nonlinear interaction terms or weapons effects. This could also lead to the exploration of the inherent nonlinearities in the Spatial Dynamics terms, which essentially describes the movement or tactics employed by a force. For example, an increase in weapon lethality (kill probability) may not necessarily have the expected corresponding increase in casualties due to the effects of the enemy's tactics. It may not be sound to assume a doubling of this probability will yield double the number of casualties.

We suggest that our continuous model be used in conjunction with agent-based wargames to act as a combined testbed for the purpose of concept exploration.

9.2 Summary

In this thesis we have derived a partial differential equation model representing combat, using biological aggregation models to more adequately represent the spatial aspects of combat and social motivational factors. The numerical methods used were based on the Method of Lines technique with extra and essential constraints of conservation of mass and positivity enforced by the appropriate use of flux limiters. Again the field of biological modelling provided the numerical framework. Comparisons to Protopopescu *et al.*'s results found similar density losses over time, however demonstrated maintenance of a constant interior density despite incurring losses through firing effects. This had not been achieved in any previous continuous combat modelling and counters the main criticism of Lanchester Equation-type models of neglecting the spatial component of combat.

This one dimensional model was then extended to two dimensions and the same process of comparisons to previous results completed. Again the maintenance of equilibrium density was demonstrated throughout the simulations. We also showed that a scenario can be viewed as a series of sub-battles characterised by periods of constant density loss. This was the intended method of implementation of the LEs as suggested by Lanchester himself. Through the addition of obstacles in the domain, we were able to show a bifurcation point induced by a force navigating around obstacle which is dependent upon the social factor of attraction. Once the separation distance of the force halves exceeds the attraction range, there is no reconstitution of the force resulting in the formation of two distinct force profiles.

Once this baseline work had been completed, comparisons to a series of ISAAC scenarios were made as was the main aim of this research. These ISAAC scenarios consisted of two homogeneous forces whose parameters remain unchanged throughout the simulations yet were deemed to exhibit adaptive behaviour. Our model was shown to provide an easier and alternate way of explaining observed behaviour rather than through the anthropomorphisation of agents. This highlighted the dangers of using atypical instances of stochastic models to drive analysis. The effects of the inherent stochasticity of ISAAC led us to induce a type of controlled stochasticity approximation through the offsetting of the initial density profiles. We were able to confirm our conjectures that many of the observed ISAAC behaviours were due to asymmetry. This also enabled us to determine the mean behaviour of an ISAAC scenario and more easily understand the effects of parameter changes.

Our model then provided a basis for the addition and exploration of other tactics through modifications to the Spatial Dynamics terms. An example of a tactic of density response to enemy detection, similar to predator avoidance models, was implemented and explored. By comparing both an expansion or contraction response with a simple baseline scenario, differences in the observed dynamics and overall density losses were found. This may also drive the further development of CA wargames through the suggestion of modifications to or alternative forms of the discrete counterparts to the interaction terms.

In conclusion we have shown that a continuum of forces is able to behave in a manner similar to a collection of individual autonomous agents, demonstrating decentralised self-organisation and adaptation of tactics to suit a variety of combat situations. This is a significant step toward developing a set of realistic continuous equations for combat modelling and provides a framework for expansion and exploration of tactics.

9.3 Future Work

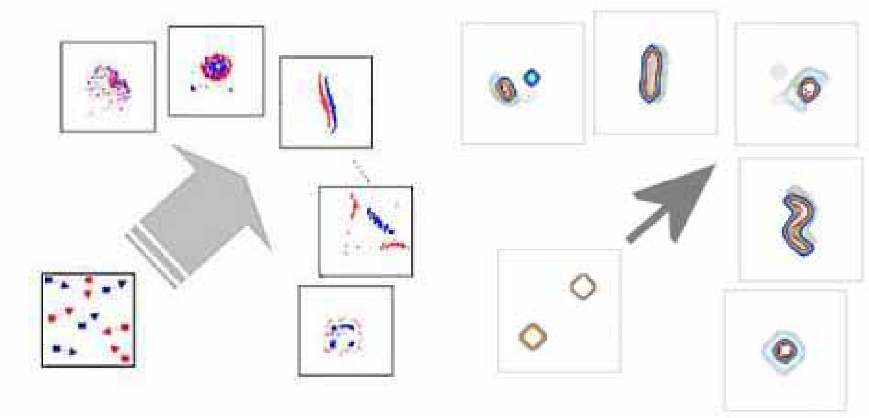
We suggest several possible areas for further research.

This research did not address other ISAAC scenarios such as LOCALCMD or GLBALCMD where communication between agents within a user-defined range affects the evolution of the scenarios. Inclusion of the local command and global command functionalities and of command personality in the penalty function (6.1) would require further significant modification to our continuous model. Addition of these comparisons will result in a more complete development of a continuous counterpart to ISAAC.

The addition of obstacles has been completed, however only force movement is restricted at present with weapons remaining unaffected. While this can be seen as a good approximation to the navigation over differing terrain types, it is not adequate for the representation of walls or buildings. ISAAC and other cellular automaton models such as the Map Aware Non-uniform Automata (MANA) model have the additional restriction on weapons capabilities through these obstacles such that they effectively act as walls. If our model was extended to allow for additional weapon restrictions of this nature, simple modelling of urban environments would then become possible. In conjunction with CA models, this would allow a more thorough exploration of urban and modern warfare modelling which is of great interest to many research organisations and the defence community.

Modifications to the Spatial Dynamics terms undertaken in Chapter 8 demonstrated the flexibility of modifying tactics and the effects this has on the observed dynamics and temporal dependency of density losses. This does raise the question however; what are the important or essential features of combat that need to be captured in the spatial dynamics? Further research may need to be conducted to ascertain the main influences on soldier behaviour in order to ensure they are reflected in modelling. Other methods of modelling social interactions could serve as a valuable resource for this.

Unlike the comparison made between an individual-based model and PDE in [10], the PDEs here were not derived from taking the limit as the number of soldiers tends to infinity. The derivation method used in [10] consisted of converting the IBM into a set of stochastic differential equations which were then converted into their corresponding PDEs. Although this approach cannot facilitate an exact comparison (as was the case for [10]) due to the stochasticity of ISAAC, the form of the resulting PDEs may differ from those developed here and therefore shed a different light on the observed dynamics. This process can also be viewed as a form of verification of ISAAC and more weight given to the assertion that the PDE results do represent the true mean behaviour of ISAAC.



(a) ISAAC

(b) Continuous

Figure 9.1: Demonstration of potential behaviours of ISAAC and Our PDE model.

Appendix A

Symbols and Abbreviations

CA	Cellular Automaton/Automata
IBM	Individual-Based Model
MOL	Method of Lines
ISAAC	Irreducible Semi-Autonomous Adaptive Combat
ISAACA	Irreducible Semi-Autonomous Adaptive Combat Agent
MANA	Map Aware Non-uniform Automata
PDE Symbols	
$u(x, y, t)$	The positive troop density of Force u at the given position and time.
$v(x, y, t)$	The positive troop density of Force v at the given position and time.
$D_i(x, y, t)$	The diffusion coefficient for Force $i(i = u, v)$ that may be spatially, temporally or otherwise dependent.
$V_i(x, y, t)$	The velocity coefficient for Force $i(i = u, v)$ that may be spatially, temporally or otherwise dependent.
A_{a_i}	The strength of the internal attraction of Force $i(i = u, v)$.
A_{r_i}	The strength of the internal repulsion of Force $i(i = u, v)$. The ratio of $\frac{A_{a_i}}{A_{r_i}}$ gives the desired internal density of Force i.
K_{a_i}	The form of the attraction kernel for Force $i(i = u, v)$. For the results presented here, the form is $K_a * u = \int_{y-r_a}^{y+r_a} \int_{x-r_a}^{x+r_a} K_a(x - X, y - Y)u(X, Y)dXdY$
K_{r_i}	The form of the repulsion kernel for Force $i(i = u, v)$. For the results presented here, the form is $K_r * u = \int_{y-r_r}^{y+r_r} \int_{x-r_r}^{x+r_r} K_r(x - X, y - Y)u(X, Y)dXdY$
r_{a_i}	The range over which the attraction kernel for Force $i(i = u, v)$ acts.
r_{r_i}	The range over which the repulsion kernel for Force $i(i = u, v)$ acts. $r_{r_i} < r_{a_i}$

k_i	The form of the area fire kernel for Force $i(i = u, v)$. For the results presented here, the form is $k_i(x, y) = \beta_i e^{-\nu_i \left\| \sqrt{(x-X)^2+(y-Y)^2} \right\|} : \mathbb{R}^2 \rightarrow \mathbb{R}^+$ for the rifle kernel, and $k(x, y) = \beta e^{-\nu \sqrt{\|((x-X)^2+(y-Y)^2)-r_{op}\|}} : \mathbb{R}^2 \rightarrow \mathbb{R}^+$ for the artillery kernel.
r_{af_i}	The range over which the area fire kernel for Force $i(i = u, v)$ acts.
r_{op_i}	The optimal range for the area artillery fire kernel for Force $i(i = u, v)$.
β_i	Variable used in the calculation of area fire losses for Force $i(i = u, v)$.
ν_i	Variable used in the calculation of area fire losses for Force $i(i = u, v)$.
d_i	The aimed fire coefficient for Force $i(i = u, v)$.
r_S	Sensor range used for the calculation of friendly and enemy forces.
Δc	Threshold number of soldiers required to trigger change in velocity calculation.
N_i	Number of soldiers of Force $i(i = u, v)$ calculated within sensor range r_S.
ISAAC Symbols	
$Z(x, y)$	Penalty function
w_1	Weighting parameter for alive friendly agents.
w_2	Weighting parameter for alive enemy agents.
w_3	Weighting parameter for injured friendly agents.
w_4	Weighting parameter for injured enemy agents.
w_5	Weighting parameter for friendly flag.
w_6	Weighting parameter for enemy flag.
r_S	Sensor range.
r_F	Fire range.
r_T	Threshold range.
r_C	Communications range.
r_M	Movement range.
d_{new}	Distance to flag (either friendly or enemy) from prospective new position.
d_{old}	Distance to flag (either friendly or enemy) from current position.
s_i	Scaling factor for ISAACs of type $i(i = red, blue)$.
N_i	Number of ISAACAs $i(i = red, blue)$ within sensor range.
Δc	Threshold number of ISAACAa required to trigger combat constraint.

Table A.1: Symbols and Abbreviations used.

Appendix B

Matlab Code, One Dimension

The input files for the one dimensional model as developed in Chapter 4.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%               Main Matlab File                               %
%               1D Model                                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear;
% --------------------- INITIALIZE VARIABLES --------------------
Dens = 10;                          % initial density
w = 20;                             % initial width of forces
N = 100;                            % number of grid points
Df = zeros(N,1); Dh = zeros(N,1);   % diffusion coefficient matrix
Dcoeff = 5;                         % default diffusion
tau = 0.0000001;                    % time step
delt = 0.000001;    % time step for firing calcs
endtime = 0.003;                    % end time of simulation
L = 1.0;                            % domain length
deltx = L/(N - 1);                  % domain discretisation
eps = 1e-30;                        % for limiter calculations

% ---------------------------- KERNELS --------------------------
a=ones(N,1)*(1:N)-(1:N)'*(ones(N,1))';  % auxiliary matrix
ae=20;                             % default convection
Aa=20;                             % attraction strength
Ar=2;                              % repulsion strength
ra=5;                              % range of attraction
rr=5;                              % range of repulsion
% odd attraction kernel
sa=0.5*sign(a).*(1-sign(abs(a)-ra-0.5));
% odd repulsion kernel
sr=0.5*sign(a).*(1-sign(abs(a)-rr-0.5));
sb=-0.5*(1-sign(abs(a)-0.5));   % for convection
sf=-ae*sb-Aa*sa; sr=Ar*sr;    % equation for force f
sh=ae*sb-Aa*sa;                  % equation for force h

% ------------------ INITIALISE FORCE PROFILES ------------------

h=zeros(N,1);             % initialise force h to zero
```

```
h_new=zeros(N,1);         % initialise force h temp matrix to zero
% initialise force profile
h(70:90) = 10*ones(21,1);

f=zeros(N,1);             % initialise force f to zero
f_new=zeros(N,1);         % initialise force f temp matrix to zero
% initialise force profile
f(10:30) = 10*ones(21,1);
% ------------------------- MAIN LOOP -------------------------
count=0;                  % counter for displaying figures
imgcount=1;               % counter for movie
rejcount=0;               % rejected step counter
currenttime=0;
stoptime=1e-3;
resultcount=2;

while currenttime<endtime

% force f
for i=1:N
if f(i)>0.1*Aa/Ar
Df(i)=Dcoeff;
else
Df(i)=Dcoeff*f(i);
end
end
% one full time step
ffull=RK(f,sf,sr,eps,N,deltx,tau);
ffull(2:N-1)=ffull(2:N-1)+...
(tau/deltx^2)*(0.5*(Df(3:N)+Df(2:N-1))...
.*(ffull(3:N)-ffull(2:N-1))-0.5*(Df(2:N-1)+Df(1:N-2))...
.*(ffull(2:N-1)-ffull(1:N-2)));;
ffull(1)=ffull(1)+(tau/deltx^2)*(0.5*(Df(2)+Df(1))...
.*(ffull(2)-ffull(1))-0.5*(Df(1)+Df(N)).*(ffull(1)-ffull(N)));
ffull(N)=ffull(N)+(tau/deltx^2)*(0.5*(Df(1)+Df(N))...
.*(ffull(1)-ffull(N))-0.5*(Df(N)+Df(N-1))...
.*(ffull(N)-ffull(N-1)));

% two half time steps
fhalf=RK(f,sf,sr,eps,N,deltx,tau/2);
fhalf(2:N-1)=fhalf(2:N-1)+((tau/2)/deltx^2)...
*(0.5*(Df(3:N)+Df(2:N-1)).*(fhalf(3:N)-fhalf(2:N-1))...
-0.5*(Df(2:N-1)+Df(1:N-2)).*(fhalf(2:N-1)-fhalf(1:N-2)));
fhalf(1)=fhalf(1)+((tau/2)/deltx^2)*(0.5*(Df(2)+Df(1))...
.*(fhalf(2)-fhalf(1))-0.5*(Df(1)+Df(N)).*(fhalf(1)-fhalf(N)));
fhalf(N)=fhalf(N)+((tau/2)/deltx^2)*(0.5*(Df(1)+Df(N))...
.*(fhalf(1)-fhalf(N))-0.5*(Df(N)+Df(N-1))...
.*(fhalf(N)-fhalf(N-1)));
for i=1:N
if fhalf(i)>0.1*Aa/Ar
Df(i)=Dcoeff;
else
Df(i)=Dcoeff*fhalf(i);
end
end
fhalf=RK(fhalf,sf,sr,eps,N,deltx,tau/2);
fhalf(2:N-1)=fhalf(2:N-1)+((tau/2)/deltx^2)...
*(0.5*(Df(3:N)+Df(2:N-1)).*(fhalf(3:N)-fhalf(2:N-1))...
-0.5*(Df(2:N-1)+Df(1:N-2)).*(fhalf(2:N-1)-fhalf(1:N-2)));
fhalf(1)=fhalf(1)+((tau/2)/deltx^2)*(0.5*(Df(2)+Df(1))...
.*(fhalf(2)-fhalf(1))-0.5*(Df(1)+Df(N)).*(fhalf(1)-fhalf(N)));
fhalf(N)=fhalf(N)+((tau/2)/deltx^2)*(0.5*(Df(1)+Df(N))...
.*(fhalf(1)-fhalf(N))-0.5*(Df(N)+Df(N-1))...
.*(fhalf(N)-fhalf(N-1)));

% tolerance checking for force f
rhof=TimeStepCheck(f,ffull,fhalf,N);

% force h
for i=1:N
if h(i)>0.1*Aa/Ar
Dh(i)=Dcoeff;
else
Dh(i)=Dcoeff*h(i);
end
end
% one full time step
hfull=RK(h,sh,sr,eps,N,deltx,tau);
hfull(2:N-1)=hfull(2:N-1)+...
(tau/deltx^2)*(0.5*(Dh(3:N)+Dh(2:N-1))...
.*(hfull(3:N)-hfull(2:N-1))-0.5*(Dh(2:N-1)+Dh(1:N-2))...
.*(hfull(2:N-1)-hfull(1:N-2)));;
hfull(1)=hfull(1)+(tau/deltx^2)*(0.5*(Dh(2)+Dh(1))...
.*(hfull(2)-hfull(1))-0.5*(Dh(1)+Dh(N)).*(hfull(1)-hfull(N)));
hfull(N)=hfull(N)+(tau/deltx^2)*(0.5*(Dh(1)+Dh(N))...
.*(hfull(1)-hfull(N))-0.5*(Dh(N)+Dh(N-1)).*(hfull(N)-hfull(N-1)));

% two half time steps
hhalf=RK(h,sh,sr,eps,N,deltx,tau/2);
hhalf(2:N-1)=hhalf(2:N-1)+((tau/2)/deltx^2)...
*(0.5*(Dh(3:N)+Dh(2:N-1)).*(hhalf(3:N)-hhalf(2:N-1))...
-0.5*(Dh(2:N-1)+Dh(1:N-2)).*(hhalf(2:N-1)-hhalf(1:N-2)));
hhalf(1)=hhalf(1)+((tau/2)/deltx^2)*(0.5*(Dh(2)+Dh(1))...
.*(hhalf(2)-hhalf(1))-0.5*(Dh(1)+Dh(N)).*(hhalf(1)-hhalf(N)));
```

```
hhalf(N)=hhalf(N)+((tau/2)/deltx^2)*(0.5*(Dh(1)+Dh(N))...
.*(hhalf(1)-hhalf(N))-0.5*(Dh(N)+Dh(N-1)).*(hhalf(N)-hhalf(N-1)));
for i=1:N
if hhalf(i)>0.1*Aa/Ar
Dh(i)=Dcoeff;
else
Dh(i)=Dcoeff*hhalf(i);
end
end
hhalf=RK(hhalf,sh,sr,eps,N,deltx,tau/2);
hhalf(2:N-1)=hhalf(2:N-1)+((tau/2)/deltx^2)...
*(0.5*(Dh(3:N)+Dh(2:N-1)).*(hhalf(3:N)-hhalf(2:N-1))...
-0.5*(Dh(2:N-1)+Dh(1:N-2)).*(hhalf(2:N-1)-hhalf(1:N-2)));
hhalf(1)=hhalf(1)+((tau/2)/deltx^2)*(0.5*(Dh(2)+Dh(1))...
.*(hhalf(2)-hhalf(1))-0.5*(Dh(1)+Dh(N)).*(hhalf(1)-hhalf(N)));
hhalf(N)=hhalf(N)+((tau/2)/deltx^2)*(0.5*(Dh(1)+Dh(N))...
.*(hhalf(1)-hhalf(N))-0.5*(Dh(N)+Dh(N-1)).*(hhalf(N)-hhalf(N-1)));

% tolerance checking for force f
rhoh=TimeStepCheck(h,hfull,hhalf,N);

tauold = tau;
rho=max(rhof,rhoh);
tau=tau*min(2,max((0.8/(nthroot(rho,3))),0.25));

if rho>1
% step rejected
'rejected'
rejcount=rejcount+1;
continue
else
% use two half step results as higher accuracy
f=fhalf;
h=hhalf;
currenttime=currenttime+tau;
if endtime-currenttime<tau
tau=endtime-currenttime;
end
count=count+1;
% calculate firing effects btween forces
[f,h]=areafire(f,h,deltx,N,tauold/delt);
if currenttime>stoptime
fresult(resultcount,:)=f;
hresult(resultcount,:)=h;
resultcount=resultcount+1;
stoptime=stoptime+1e-3;
end
```

```
end

if mod(count,10)==0
w=figure(1);
clf
plot((1:N),f,'b',(1:N),h,'r --')
axis([0 N 0 11])
ylabel('Troop Density')
xlabel('x')
title('Area Fire')
F(imgcount)=getframe(w);
imgcount=imgcount+1;
sum(f)
end
end

% ------------------ OUTPUTS, FIGURES & MOVIES ------------------
['Rejected Steps:' int2str(rejcount)]

movie2avi(F,'1DAreaFire.avi');

% snapshots of simulation
w=figure(1);
plot((1:N),fresult(1,:),'b',(1:N),fresult(2,:),'b --',(1:N),...
fresult(3,:),'b -.',(1:N),f,'b :',(1:N),hresult(1,:),'r',...
(1:N),hresult(2,:),'r --',(1:N),hresult(3,:),'r -.',(1:N),h,'r :')
axis([0 N 0 11]);
ylabel('Troop Density')
xlabel('x')
title('Area Fire, Density Dependent Diffusion')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%               Runge Kutta                                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [f_new]=RK(f,s,sr,eps,N,deltx,h);
% Runge Kutta scheme
flux=fluxsub(f,s,sr,eps,N);
k1(2:N,1)=(-1/deltx).*(flux(2:N)-flux(1:N-1));
k1(1,1)=(-1/deltx).*(flux(1)-flux(N));

flux = fluxsub(f+h/2.*k1,s,sr,eps,N);
k2(2:N,1)=(-1/deltx).*(flux(2:N)-flux(1:N-1));
k2(1,1)=(-1/deltx).*(flux(1)-flux(N));

flux=fluxsub(f+h/2.*k1+h/2.*k2,s,sr,eps,N);
k3(2:N,1)=(-1/deltx).*(flux(2:N)-flux(1:N-1));
```

```
k3(1,1)=(-1/deltx).*(flux(1)-flux(N));

f_new(1:N,1)=f(1:N)+(h/3)*(k1+k2+k3);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%               Flux and Limiter Calculation                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [flux]=fluxsub(f,s,sr,eps,N)
g=zeros(N,1);
speed=zeros(N,1);
r=zeros(N,1);
theta=zeros(N,1);
flux=zeros(N,1);

% calculate convolutions and fluxes
g(1:N)=(f'*s)+f'.*(f'*sr);
speed=g;
g=g.*f;

% calculate ratio of fuxes about each point
r(2:N-1,1)=(g(3:N)-g(2:N-1)+eps)./(g(2:N-1)-g(1:N-2)+eps);
r(1,1)=(g(2)-g(1)+eps)./(g(1)-g(N)+eps);
r(N,1)=(g(1)-g(N)+eps)./(g(N)-g(N-1)+eps);

% FLUX LIMITERS
% van Leer limiter
theta(1:N,1)=(r+abs(r))./(1+abs(r));
thetainv(1:N,1)=(r.^(-1)+abs(r.^(-1)))./(1+abs(r.^(-1)));

% First-Order Upwind
% theta=zeros(N,1);
% thetainv=zeros(N,1);

% Second-Order Central
% delta=2;
% theta=max(0,min(r,delta));
% thetainv=max(0,min(r.^(-1),delta));

% Second-Order Upwind
% delta=2;
% theta=max(0,min(2*r,min(delta,1)));
% thetainv=max(0,min(2*r.^(-1),min(delta,1)));

% enforce flux limiters based on direction of speed
for i=1:N
```

```
if f(i)>0
if speed(i)>0
if i==1
flux(1,1)=(g(1)+0.5.*theta(1).*(g(1)-g(N)));
else
flux(i,1)=(g(i)+0.5.*theta(i).*(g(i)-g(i-1)));
end
elseif speed(i)<0
if i==1
temp=(g(1)+0.5.*thetainv(1).*(g(1)-g(2)));
flux(N,1)=flux(N,1)+temp;
elseif i==N
temp=(g(N)+0.5.*thetainv(N).*(g(N)-g(1)));
flux(N-1,1)=flux(N-1,1)+temp;
else
temp=(g(i)+0.5.*thetainv(i).*(g(i)-g(i+1)));
flux(i-1,1)=flux(i-1,1)+temp;
end
else
flux(i,1)=0;
end
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%               Firing Effects                                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [f,h]=areafire(f,h,deltx,N,tau);
v1=0.8;
v2=0.8;
beta11=0.00008;
gamma11=0.00008;
beta12=0;
gamma12=0;

lhs1=zeros(N,1); lhs2 = zeros(N,1);

a=ones(N,1)*(1:N)-(1:N)'*(ones(N,1))';
areafiref(1:N,1)=-deltx*gamma11*f.*(h'*exp(-v1.*abs(a*deltx)))';
aimedfiref(1:N,1)=-gamma12.*f.*h;
lhs1=areafiref+aimedfiref;

areafireh(1:N,1)=-deltx*beta11*h.*(f'*exp(-v2.*abs(a*deltx)))';
aimedfireh(1:N,1)=-beta12.*f.*h;
lhs2=areafireh+aimedfireh;
```

```
% enforce losses
f(1:N) = f(1:N) + lhs1*tau;
h(1:N) = h(1:N) + lhs2*tau;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%               Tolerance Check                                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [rho]=TimeStepCheck(f,ffull,fhalf,N);
atol=1e-6;
rtol=1e-6;

rho=sum(((fhalf-ffull)./(atol+rtol.*abs(f))).^2);
rho=(sqrt(rho/N))/3;
end
```

Appendix C

Matlab Code, Two Dimensions

The input files for the two dimensional model as developed in Chapter 5.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%               Main Matlab File                                %
%               2D Model                                        %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear;
warning off all
% --------------------- INITIALIZE VARIABLES --------------------
N=50;               % number of grid points along both axes
Lx=1.0; Ly=Lx;      % domain length, width
deltx=Lx/(100-1);   % domain discretisation
delty=deltx;
endtime=0.001;      % end time of simulation
tau=1e-7;           % time step
Df=zeros(N,N);      % diffusion matricies
Dh=zeros(N,N);
Dcoeff=5;           % default diffusion coefficient
lossesf=[0];        % losses due to firing effects, force f
lossesh=[0];        % losses due to firing effects, force h
timevec=[0];

% ------------------ INITIALISE FORCE PROFILES ------------------
f=zeros(N,N);       % initialise force f to zero
h=zeros(N,N);       % initialise force h to zero
ffull=zeros(N,N);   % initialise force f temp matricies to zero
fhalf=zeros(N,N);
hfull=zeros(N,N);   % initialise force h temp matricies to zero
hhalf=zeros(N,N);
[X,Y]=meshgrid(1:N,1:N);

% initialise force f profile
rho=5; mux=18; muy=15;
f=5/(rho*sqrt(2*pi))^0.5*exp(-((X-mux).^2+(Y-muy).^2)./(2*rho^2));
f=(f>1).*8;

% initialise force h profile
rho=5; mux=32; muy=35;
h=5/(rho*sqrt(2*pi))^0.5*exp(-((X-mux).^2+(Y-muy).^2)./(2*rho^2));
```

```
h=(h>1).*8;

% initialise convection, attraction, repulsion strengths and ranges
faey=60*ones(N,N); fAay=5; fAry=0.5; fray=5; frry=5;
faex=60*ones(N,N); fAax=5; fArx=0.5; frax=5; frrx=5;
haey=-60*ones(N,N); hAay=5; hAry=0.5; hray=5; hrry=5;
haex=-60*ones(N,N); hAax=5; hArx=0.5; hrax=5; hrrx=5;
% ranges and thresholds for attack metapersonality
frc=5; fNc=1000;
hrc=5; hNc=1000;
% type of attack metapersonality
fattack=-1;
hattack=-1;

% -------------------- CONVOLUTION KERNELS ----------------------
M=2*N-1;
a=ones(M,1)*(1:M)-(1:M)’*(ones(M,1))’;
d=zeros(M,M);

for i=1:M
d(i,:)=sqrt(a(i,(M+1)/2).^2+a((M+1)/2,:).^2);
end

% attraction kernels for force f and h
fdax=(d<=frax)*1; fdax(1:N,:)=-1*fdax(1:N,:);
fdax(N,:)=0; fdax=fdax(N-frax:N+frax,N-fray:N+fray);
hdax=(d<=hrax)*1; hdax(1:N,:)=-1*hdax(1:N,:);
hdax(N,:)=0; hdax=hdax(N-hrax:N+hrax,N-hray:N+hray);

% repulsion kernels for force f and h
fdrx=(d<=frrx)*1; fdrx(1:N,:)=-1*fdrx(1:N,:);
fdrx(N,:)=0; fdrx=fdrx(N-frrx:N+frrx,N-frry:N+frry);
hdrx=(d<=hrrx)*1; hdrx(1:N,:)=-1*hdrx(1:N,:);
hdrx(N,:)=0; hdrx=hdrx(N-hrrx:N+hrrx,N-hrry:N+hrry);

% kernels for calculating friendlies and enemies
fdtr=(d<=frc)*1; fdtr=fdtr(N-frc:N+frc,N-frc:N+frc);
hdtr=(d<=hrc)*1; hdtr=hdtr(N-hrc:N+hrc,N-hrc:N+hrc);

% kernels for calculating friendly and enemy centres of mass
fdcx=(d<=frc)*1; fdcx(1:N,:)=-1*fdcx(1:N,:); fdcx(N,N)=0;
fdcx(N,:)=0; fdcx=fdcx(N-frc:N+frc,N-frc:N+frc);
hdcx=(d<=hrc)*1; hdcx(1:N,:)=1*hdcx(1:N,:); hdcx(N,N)=0;
hdcx(N,:)=0; hdcx=hdcx(N-hrc:N+hrc,N-hrc:N+hrc);

% -------------------- INITIALISE GRAPHING ----------------------
initialnumberf=sum(sum(f));
```

```
initialnumberh=sum(sum(h));
step=1;
% main figure for movie
w=figure(1);
tripleplot(w,f,h,0);
F(1)=getframe(w);
% 3D snapshots
w=figure(2);
clf
h1=subplot(3,4,1);
subplotRB3D(h1,f,h,0);
% 2D snapshots
w=figure(3);
clf
h1=subplot(3,4,1);
subplotRB2D(h1,f,h,0);

% -------------------------- MAIN LOOP --------------------------
count=0;                  % counter for displaying figures
rejcount=0;               % counter for rejected steps
imgcount=2;               % counter for movie
currenttime=0;
t0=clock;
stophere=endtime/10; % 10 snapshots per run
stopcount=2;

while currenttime < endtime

% keep a copy in case step rejected
f_old=f; h_old=h;

% Force f

% one full time step
Df=(f>0.1*fAax/fArx).*Dcoeff+(f<=0.1*fAax/fArx).*Dcoeff.*f;

ffull(2:N-1,2:N-1)=ffull(2:N-1,2:N-1)+(tau/deltx^2)*...
(0.5*(Df(3:N,2:N-1)+Df(2:N-1,2:N-1)).*(ffull(3:N,2:N-1)-...
ffull(2:N-1,2:N-1))-0.5*(Df(2:N-1,2:N-1)+Df(1:N-2,2:N-1)).*...
(ffull(2:N-1,2:N-1)-ffull(1:N-2,2:N-1)))+...
(tau/delty^2)*(0.5*(Df(2:N-1,3:N)+Df(2:N-1,2:N-1)).*...
(ffull(2:N-1,3:N)-ffull(2:N-1,2:N-1))-0.5*(Df(2:N-1,2:N-1)+...
Df(2:N-1,1:N-2)).*(ffull(2:N-1,2:N-1)-ffull(2:N-1,1:N-2)));
ffull(1,1:N)=ffull(1,1:N)+(tau/deltx^2)*...
(0.5*(Df(2,1:N)+Df(1,1:N)).*(ffull(2,1:N)-ffull(1,1:N))-...
0.5*(Df(1,1:N)+Df(N,1:N)).*(ffull(1,1:N)-ffull(N,1:N)));
ffull(N,1:N)=ffull(N,1:N)+(tau/deltx^2)*...
```

```
(0.5*(Df(1,1:N)+Df(N,1:N)).*(ffull(1,1:N)-ffull(N,1:N))-...
0.5*(Df(N,1:N)+Df(N-1,1:N)).*(ffull(N,1:N)-ffull(N-1,1:N)));
ffull(1:N,1)=ffull(1:N,1)+(tau/delty^2)*...
(0.5*(Df(1:N,2)+Df(1:N,1)).*(ffull(1:N,2)-ffull(1:N,1))-...
0.5*(Df(1:N,1)+Df(1:N,N)).*(ffull(1:N,1)-ffull(1:N,N)));
ffull(1:N,N)=ffull(1:N,N)+(tau/delty^2)*...
(0.5*(Df(1:N,1)+Df(1:N,N)).*(ffull(1:N,1)-ffull(1:N,N))-...
0.5*(Df(1:N,N)+Df(1:N,N-1)).*(ffull(1:N,N)-ffull(1:N,N-1)));
ffull=RK2D(f,h,N,deltx,delty,tau,-fAax*fdax,fArx*fdrx,faex,...
faey,frax,frrx,fdcx,fdtr,fNc,fattack);

% two half steps
fhalf(2:N-1,2:N-1)=fhalf(2:N-1,2:N-1)+((tau/2)/deltx^2)*...
(0.5*(Df(3:N,2:N-1)+Df(2:N-1,2:N-1)).*(fhalf(3:N,2:N-1)-...
fhalf(2:N-1,2:N-1))-0.5*(Df(2:N-1,2:N-1)+Df(1:N-2,2:N-1)).*...
(fhalf(2:N-1,2:N-1)-fhalf(1:N-2,2:N-1)))+((tau/2)/delty^2)*...
(0.5*(Df(2:N-1,3:N)+Df(2:N-1,2:N-1)).*(fhalf(2:N-1,3:N)-...
fhalf(2:N-1,2:N-1))-0.5*(Df(2:N-1,2:N-1)+Df(2:N-1,1:N-2)).*...
(fhalf(2:N-1,2:N-1)-fhalf(2:N-1,1:N-2)));
fhalf(1,1:N)=fhalf(1,1:N)+(tau/deltx^2)*...
(0.5*(Df(2,1:N)+Df(1,1:N)).*(fhalf(2,1:N)-fhalf(1,1:N))-...
0.5*(Df(1,1:N)+Df(N,1:N)).*(fhalf(1,1:N)-fhalf(N,1:N)));
fhalf(N,1:N)=fhalf(N,1:N)+(tau/deltx^2)*....
(0.5*(Df(1,1:N)+Df(N,1:N)).*(fhalf(1,1:N)-fhalf(N,1:N))-...
0.5*(Df(N,1:N)+Df(N-1,1:N)).*(fhalf(N,1:N)-fhalf(N-1,1:N)));
fhalf(1:N,1)=fhalf(1:N,1)+(tau/delty^2)*...
(0.5*(Df(1:N,2)+Df(1:N,1)).*(fhalf(1:N,2)-fhalf(1:N,1))-...
0.5*(Df(1:N,1)+Df(1:N,N)).*(fhalf(1:N,1)-fhalf(1:N,N)));
fhalf(1:N,N)=fhalf(1:N,N)+(tau/delty^2)*...
(0.5*(Df(1:N,1)+Df(1:N,N)).*(fhalf(1:N,1)-fhalf(1:N,N))-...
0.5*(Df(1:N,N)+Df(1:N,N-1)).*(fhalf(1:N,N)-fhalf(1:N,N-1)));
fhalf=RK2D(f,h,N,deltx,delty,tau/2,-fAax*fdax,fArx*fdrx,faex,...
faey,frax,frrx,fdcx,fdtr,fNc,fattack);

Df=(fhalf>0.1*fAax/fArx).*Dcoeff+...
(fhalf<=0.1*fAax/fArx).*Dcoeff.*fhalf;

fhalf(2:N-1,2:N-1)=fhalf(2:N-1,2:N-1)+((tau/2)/deltx^2)*...
(0.5*(Df(3:N,2:N-1)+Df(2:N-1,2:N-1)).*(fhalf(3:N,2:N-1)-...
   fhalf(2:N-1,2:N-1))-0.5*(Df(2:N-1,2:N-1)+Df(1:N-2,2:N-1)).*...
(fhalf(2:N-1,2:N-1)-fhalf(1:N-2,2:N-1)))+((tau/2)/delty^2)*...
(0.5*(Df(2:N-1,3:N)+Df(2:N-1,2:N-1)).*(fhalf(2:N-1,3:N)-...
fhalf(2:N-1,2:N-1))-0.5*(Df(2:N-1,2:N-1)+Df(2:N-1,1:N-2)).*...
(fhalf(2:N-1,2:N-1)-fhalf(2:N-1,1:N-2)));
fhalf(1,1:N)=fhalf(1,1:N)+((tau/2)/deltx^2)*...
(0.5*(Df(2,1:N)+Df(1,1:N)).*(fhalf(2,1:N)-fhalf(1,1:N))-...
0.5*(Df(1,1:N)+Df(N,1:N)).*(fhalf(1,1:N)-fhalf(N,1:N)));
```

```
fhalf(N,1:N)=fhalf(N,1:N)+((tau/2)/deltx^2)*...
(0.5*(Df(1,1:N)+Df(N,1:N)).*(fhalf(1,1:N)-fhalf(N,1:N))-...
0.5*(Df(N,1:N)+Df(N-1,1:N)).*(fhalf(N,1:N)-fhalf(N-1,1:N)));
fhalf(1:N,1)=fhalf(1:N,1)+((tau/2)/delty^2)*...
(0.5*(Df(1:N,2)+Df(1:N,1)).*(fhalf(1:N,2)-fhalf(1:N,1))-...
0.5*(Df(1:N,1)+Df(1:N,N)).*(fhalf(1:N,1)-fhalf(1:N,N)));
fhalf(1:N,N)=fhalf(1:N,N)+((tau/2)/delty^2)*...
(0.5*(Df(1:N,1)+Df(1:N,N)).*(fhalf(1:N,1)-fhalf(1:N,N))-...
0.5*(Df(1:N,N)+Df(1:N,N-1)).*(fhalf(1:N,N)-fhalf(1:N,N-1)));
fhalf=RK2D(fhalf,h,N,deltx,delty,tau/2,-fAax*fdax,fArx*fdrx,...
faex,faey,frax,frrx,fdcx,fdtr,fNc,fattack);

% tolerance checking for Force f
rhof=TimeStepCheck2D(f,ffull,fhalf,N,N);

% Force h

% one full time step
Dh=(h>0.1*hAax/hArx).*Dcoeff+(h<=0.1*hAax/hArx).*Dcoeff.*h;

hfull(2:N-1,2:N-1)=hfull(2:N-1,2:N-1)+(tau/deltx^2)*...
(0.5*(Dh(3:N,2:N-1)+Dh(2:N-1,2:N-1)).*(hfull(3:N,2:N-1)-...
hfull(2:N-1,2:N-1))-0.5*(Dh(2:N-1,2:N-1)+Dh(1:N-2,2:N-1)).*...
(hfull(2:N-1,2:N-1)-h(1:N-2,2:N-1)))+(tau/delty^2)*...
(0.5*(Dh(2:N-1,3:N)+Dh(2:N-1,2:N-1)).*(hfull(2:N-1,3:N)-...
hfull(2:N-1,2:N-1))-0.5*(Dh(2:N-1,2:N-1)+Dh(2:N-1,1:N-2)).*...
(hfull(2:N-1,2:N-1)-hfull(2:N-1,1:N-2)));
hfull(1,1:N)=hfull(1,1:N)+(tau/deltx^2)*...
(0.5*(Dh(2,1:N)+Dh(1,1:N)).*(hfull(2,1:N)-hfull(1,1:N))-...
0.5*(Dh(1,1:N)+Dh(N,1:N)).*(hfull(1,1:N)-hfull(N,1:N)));
hfull(N,1:N)=hfull(N,1:N)+(tau/deltx^2)*...
(0.5*(Dh(1,1:N)+Dh(N,1:N)).*(hfull(1,1:N)-hfull(N,1:N))-...
0.5*(Dh(N,1:N)+Dh(N-1,1:N)).*(hfull(N,1:N)-hfull(N-1,1:N)));
hfull(1:N,1)=hfull(1:N,1)+(tau/delty^2)*...
(0.5*(Dh(1:N,2)+Dh(1:N,1)).*(hfull(1:N,2)-hfull(1:N,1))-...
0.5*(Dh(1:N,1)+Dh(1:N,N)).*(hfull(1:N,1)-hfull(1:N,N)));
hfull(1:N,N)=hfull(1:N,N)+(tau/delty^2)*...
(0.5*(Dh(1:N,1)+Dh(1:N,N)).*(hfull(1:N,1)-hfull(1:N,N))-...
0.5*(Dh(1:N,N)+Dh(1:N,N-1)).*(hfull(1:N,N)-hfull(1:N,N-1)));
hfull=RK2D(h,f,N,deltx,delty,tau,-hAax*hdax,hArx*hdrx,haex,...
haey,hrax,hrrx,hdcx,hdtr,hNc,hattack);

% two half steps
hhalf(2:N-1,2:N-1)=hhalf(2:N-1,2:N-1)+((tau/2)/deltx^2)*...
(0.5*(Dh(3:N,2:N-1)+Dh(2:N-1,2:N-1)).*(hhalf(3:N,2:N-1)-...
hhalf(2:N-1,2:N-1))-0.5*(Dh(2:N-1,2:N-1)+Dh(1:N-2,2:N-1)).*...
(hhalf(2:N-1,2:N-1)-hhalf(1:N-2,2:N-1)))+((tau/2)/delty^2)*...
```

```
(0.5*(Dh(2:N-1,3:N)+Dh(2:N-1,2:N-1)).*(hhalf(2:N-1,3:N)-...
hhalf(2:N-1,2:N-1))-0.5*(Dh(2:N-1,2:N-1)+Dh(2:N-1,1:N-2)).*...
(hhalf(2:N-1,2:N-1)-hhalf(2:N-1,1:N-2)));
hhalf(1,1:N)=hhalf(1,1:N)+(tau/deltx^2)*...
(0.5*(Dh(2,1:N)+Dh(1,1:N)).*(hhalf(2,1:N)-hhalf(1,1:N))-...
0.5*(Dh(1,1:N)+Dh(N,1:N)).*(hhalf(1,1:N)-hhalf(N,1:N)));
hhalf(N,1:N)=hhalf(N,1:N)+(tau/deltx^2)*...
(0.5*(Dh(1,1:N)+Dh(N,1:N)).*(hhalf(1,1:N)-hhalf(N,1:N))-...
0.5*(Dh(N,1:N)+Dh(N-1,1:N)).*(hhalf(N,1:N)-hhalf(N-1,1:N)));
hhalf(1:N,1)=hhalf(1:N,1)+(tau/delty^2)*...
(0.5*(Dh(1:N,2)+Dh(1:N,1)).*(hhalf(1:N,2)-hhalf(1:N,1))-...
0.5*(Dh(1:N,1)+Dh(1:N,N)).*(hhalf(1:N,1)-hhalf(1:N,N)));
hhalf(1:N,N)=hhalf(1:N,N)+(tau/delty^2)*...
(0.5*(Dh(1:N,1)+Dh(1:N,N)).*(hhalf(1:N,1)-hhalf(1:N,N))-...
0.5*(Dh(1:N,N)+Dh(1:N,N-1)).*(hhalf(1:N,N)-hhalf(1:N,N-1)));
hhalf=RK2D(h,f,N,deltx,delty,tau/2,-hAax*hdax,hArx*hdrx,haex,...
haey,hrax,hrrx,hdcx,hdtr,hNc,hattack);

Dh=(hhalf>0.1*hAax/hArx).*Dcoeff+...
(hhalf<=0.1*hAax/hArx).*Dcoeff.*hhalf;

hhalf(2:N-1,2:N-1)=hhalf(2:N-1,2:N-1)+((tau/2)/deltx^2)*...
(0.5*(Dh(3:N,2:N-1)+Dh(2:N-1,2:N-1)).*(hhalf(3:N,2:N-1)-...
hhalf(2:N-1,2:N-1))-0.5*(Dh(2:N-1,2:N-1)+Dh(1:N-2,2:N-1)).*...
(hhalf(2:N-1,2:N-1)-hhalf(1:N-2,2:N-1)))+((tau/2)/delty^2)*...
(0.5*(Dh(2:N-1,3:N)+Dh(2:N-1,2:N-1)).*(hhalf(2:N-1,3:N)-...
hhalf(2:N-1,2:N-1))-0.5*(Dh(2:N-1,2:N-1)+Dh(2:N-1,1:N-2)).*...
(hhalf(2:N-1,2:N-1)-hhalf(2:N-1,1:N-2)));
hhalf(1,1:N)=hhalf(1,1:N)+((tau/2)/deltx^2)*...
(0.5*(Dh(2,1:N)+Dh(1,1:N)).*(hhalf(2,1:N)-hhalf(1,1:N))-...
0.5*(Dh(1,1:N)+Dh(N,1:N)).*(hhalf(1,1:N)-hhalf(N,1:N)));
hhalf(N,1:N)=hhalf(N,1:N)+((tau/2)/deltx^2)*...
(0.5*(Dh(1,1:N)+Dh(N,1:N)).*(hhalf(1,1:N)-hhalf(N,1:N))-...
0.5*(Dh(N,1:N)+Dh(N-1,1:N)).*(hhalf(N,1:N)-hhalf(N-1,1:N)));
hhalf(1:N,1)=hhalf(1:N,1)+((tau/2)/delty^2)*...
(0.5*(Dh(1:N,2)+Dh(1:N,1)).*(hhalf(1:N,2)-hhalf(1:N,1))-...
0.5*(Dh(1:N,1)+Dh(1:N,N)).*(hhalf(1:N,1)-hhalf(1:N,N)));
hhalf(1:N,N)=hhalf(1:N,N)+((tau/2)/delty^2)*...
(0.5*(Dh(1:N,1)+Dh(1:N,N)).*(hhalf(1:N,1)-hhalf(1:N,N))-...
0.5*(Dh(1:N,N)+Dh(1:N,N-1)).*(hhalf(1:N,N)-hhalf(1:N,N-1)));
hhalf=RK2D(hhalf,f,N,deltx,delty,tau/2,-hAax*hdax,hArx*hdrx,...
haex,haey,hrax,hrrx,hdcx,hdtr,hNc,hattack);

% tolerance checking for force h
rhoh=TimeStepCheck2D(h,hfull,hhalf,N,N);

%----------------------------------------------------------------
```

```
tauold=tau;
rho=max(rhof,rhoh);
tau=tau*min(2,max((0.8/(nthroot(rho,3))),0.25));

if rho>1 | isnan(rho)==1
% step rejected
['rejected:' int2str(rejcount)]
rejcount=rejcount+1;
f=f_old; h=h_old;
continue
else
count=count+1;
f=fhalf;
h=hhalf;
currenttime=currenttime+tau;
% check if coming toward end of simulation and adjust tau
% if necessary
if endtime-currenttime<tau
tau=endtime-currenttime;
end
timevec(count)=currenttime;
lossesf(count)=sum(sum(f));
lossesh(count)=sum(sum(h));
% calculate firing effects btween forces
[f,h,areafiref,aimedfiref,areafireh,aimedfireh]=...
areafire2D(f,h,deltx,delty,N,N,d,tauold/delt);
end

% capture figure for movie
if mod(count,40)==0
w=figure(1);
tripleplot(w,f,h,currenttime);
F(imgcount)=getframe(w);
imgcount=imgcount+1;
currenttime
['f:' int2str(sum(sum(f))) ', h:' int2str(sum(sum(h)))]
end

if currenttime>stophere
w=figure(2);
h1=subplot(3,4,stopcount);
subplotRB3D(h1,f,h,currenttime);
w=figure(3);
h1=subplot(3,4,stopcount);
subplotRB2D(h1,f,h,currenttime);
stophere=stophere+endtime/10;
stopcount=stopcount+1;
```

```
end
end

% ------------------ OUTPUTS, FIGURES & MOVIES ------------------
[’Initial F:’ int2str(initialnumberf) ’,...
Initial H:’ int2str(initialnumberh)]
[’Final F:’ int2str(sum(sum(f))) ’, Final H:’ int2str(sum(sum(h)))]
rejcount

w=figure(2);
saveas(w,’EinAll3D.fig’)
saveas(w,’EinAll3D.eps’, ’psc2’)

w=figure(3);
saveas(w,’EinAll2D.fig’)
saveas(w,’EinAll2D.eps’, ’psc2’)

w=figure(4);
clf
plot(timevec,lossesf,’r --’,’LineWidth’,2,’DisplayName’,’Force 1’)
hold all
plot(timevec,lossesh,’b :’,’LineWidth’,2,’DisplayName’,’Force 2’)
legend(’show’)
xlabel(’Time’)
ylabel(’Total Density’)
saveas(w,’losses.fig’)
saveas(w,’losses.eps’, ’psc2’)

movie2avi(F,’Ein.avi’);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%               Runge Kutta                                      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [f_new]=RK2D(f,h,N,deltx,delty,tau,da,dr,aex,aey,...
rax,rrx,dcx,dtr,Nc,attack);

% Runge_Kutta
[fluxx,fluxy]=flux2Dsub(f,h,N,da,dr,aex,aey,rax,...
rrx,dcx,dtr,Nc,attack);
k1x(2:N,:)=(-1/deltx).*(fluxx(2:N,:)-fluxx(1:N-1,:));
k1x(1,:)=(-1/deltx).*(fluxx(1,:)-fluxx(N,:));
k1y(:,2:N)=(-1/delty).*(fluxy(:,2:N)-fluxy(:,1:N-1));
k1y(:,1)=(-1/delty).*(fluxy(:,1)-fluxy(:,N));

[fluxx,fluxy]=flux2Dsub(f+tau/2.*k1x+tau/2.*k1y,h,N,da,...
dr,aex,aey,rax,rrx,dcx,dtr,Nc,attack);
```

```
k2x(2:N,:)=(-1/deltx).*(fluxx(2:N,:)-fluxx(1:N-1,:));
k2x(1,:)=(-1/deltx).*(fluxx(1,:)-fluxx(N,:));
k2y(:,2:N)=(-1/delty).*(fluxy(:,2:N)-fluxy(:,1:N-1));
k2y(:,1)=(-1/delty).*(fluxy(:,1)-fluxy(:,N));

[fluxx,fluxy]=flux2Dsub(f+tau/2.*k1x+tau/2.*k1y+tau/2.*...
k2x+tau/2.*k2y,h,N,da,dr,aex,aey,rax,rrx,dcx,dtr,Nc,attack);
k3x(2:N,:)=(-1/deltx).*(fluxx(2:N,:)-fluxx(1:N-1,:));
k3x(1,:)=(-1/deltx).*(fluxx(1,:)-fluxx(N,:));
k3y(:,2:N)=(-1/delty).*(fluxy(:,2:N)-fluxy(:,1:N-1));
k3y(:,1)=(-1/delty).*(fluxy(:,1)-fluxy(:,N));

f_new=f+(tau/3)*(k1x+k2x+k3x)+(tau/3)*(k1y+k2y+k3y);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%               Flux and Limiter Calculation                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [fluxx,fluxy]=flux2Dsub(f,h,N,dax,drx,aex,aey,...
rax,rrx,dcx,dtr,Nc,attack);

eps=1e-30;
gx=zeros(N,N); gy=zeros(N,N);
speedx=zeros(N,N); speedy=zeros(N,N);
rx=zeros(N,N); ry=zeros(N,N);
thetax=zeros(N,N); thetay=zeros(N,N);
fluxx=zeros(N,N); fluxy=zeros(N,N);
day=dax'; dry=drx'; dcy=dcx';

% -------------------------------------------------------------
% For including metapersonality approximation

numhx=conv2(h,-dcx,'same');
numhy=conv2(h,-dcy,'same');

friendly=conv2(f,dtr,'same');
enemy=conv2(h,dtr,'same');

testx=(enemy>0.5).*(((friendly-enemy)>=Nc).*numhx+...
((friendly-enemy)<Nc).*numhx*attack);
testy=(enemy>0.5).*(((friendly-enemy)>=Nc).*numhy+...
((friendly-enemy)<Nc).*numhy*attack);

attrepx=conv2(f,dax,'same')+f.*conv2(f,drx,'same');
attrepy=conv2(f,day,'same')+f.*conv2(f,dry,'same');
```

```
speedx=aex+testx+attrepx;
speedy=aey+testy+attrepy;
% -------------------------------------------------------------

% -------------------------------------------------------------
% For simple movement throughout domain
%
% speedx = aex + conv2(f,dax,’same’) + f.*conv2(f,drx,’same’);
% speedy = aey + conv2(f,day,’same’) + f.*conv2(f,dry,’same’);
% -------------------------------------------------------------

gx=speedx.*f;
gy=speedy.*f;

rx(2:N-1,:)=(gx(3:N,:)-gx(2:N-1,:)+eps)./...
(gx(2:N-1,:)-gx(1:N-2,:)+eps);
rx(1,:)=(gx(2,:)-gx(1,:)+eps)./(gx(1,:)-gx(N,:)+eps);
rx(N,:)=(gx(1,:)-gx(N,:)+eps)./(gx(N,:)-gx(N-1,:)+eps);
ry(:,2:N-1)=(gy(:,3:N)-gy(:,2:N-1)+eps)./...
(gy(:,2:N-1)-gy(:,1:N-2)+eps);
ry(:,1)=(gy(:,2)-gy(:,1)+eps)./(gy(:,1)-gy(:,N)+eps);
ry(:,N)=(gy(:,1)-gy(:,N)+eps)./(gy(:,N)-gy(:,N-1)+eps);

thetax=(rx+abs(rx))./(1+abs(rx));
thetaxinv=(rx.^(-1)+abs(rx.^(-1)))./(1+abs(rx.^(-1)));
thetay=(ry+abs(ry))./(1+abs(ry));
thetayinv=(ry.^(-1)+abs(ry.^(-1)))./(1+abs(ry.^(-1)));

for i=1:N
for j=1:N
if f(i,j)>0
if speedx(i,j)>0
if i==1
fluxx(1,j)=(gx(1,j)+0.5.*thetax(1,j).*(gx(1,j)-gx(N,j)));
else
fluxx(i,j)=(gx(i,j)+...
0.5.*thetax(i,j).*(gx(i,j)-gx(i-1,j)));
end
elseif speedx(i,j)<0
if i==1
temp=(gx(1,j)+...
0.5.*thetaxinv(1,j).*(gx(1,j)-gx(2,j)));
fluxx(N,j)=fluxx(N,j)+temp;
elseif i==N
temp=(gx(N,j)+...
0.5.*thetaxinv(N,j).*(gx(N,j)-gx(1,j)));
fluxx(N-1,j)=fluxx(N-1,j)+temp;
```

```
else
temp=(gx(i,j)+...
0.5.*thetaxinv(i,j).*(gx(i,j)-gx(i+1,j)));
fluxx(i-1,j)=fluxx(i-1,j)+temp;
end
else
fluxx(i,j)=0;
end

if speedy(i,j)>0
if j==1
fluxy(i,1)=(gy(i,1)+...
0.5.*thetay(i,1).*(gy(i,1)-gy(i,N)));
else
fluxy(i,j)=(gy(i,j)+...
0.5.*thetay(i,j).*(gy(i,j)-gy(i,j-1)));
end
elseif speedy(i,j)<0
if j==1
temp=(gy(i,1)+...
0.5.*thetayinv(i,1).*(gy(i,1)-gy(i,2)));
fluxy(i,N)=fluxy(i,N)+temp;
elseif j==N
temp=(gy(i,N)+...
0.5.*thetayinv(i,N).*(gy(i,N)-gy(i,1)));
fluxy(i,N-1)=fluxy(i,N-1)+temp;
else
temp=(gy(i,j)+...
0.5.*thetayinv(i,j).*(gy(i,j)-gy(i,j+1)));
fluxy(i,j-1)=fluxy(i,j-1)+temp;
end
else
fluxy(i,j)=0;
end
end
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%               Tolerance Check                                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [rho]=TimeStepCheck2D(f,ffull,fhalf,N,M);
atol=1e-3;
rtol=1e-3;
```

```
rho=sum(sum(((fhalf-ffull)./(atol+rtol.*abs(f))).^2));
rho=(sqrt(rho/(N*M)))/3;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%               Firing Effects                                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [f,h,areafiref,aimedfiref,areafireh,aimedfireh] = ...
areafire2D(f,h,deltx,delty,N,M,d,tau);

gammaf2=0;                % aimed fire constants
vf=0;                     % area fire constants
gammaf1=0;                % area fire constants

betah2=0;                 % aimed fire constants
vh=0;                     % area fire constants
betah1=0;                 % area fire constants

%optimalrange=15;

areafiref=zeros(N,M); aimedfiref=zeros(N,M);
areafireh=zeros(N,M); aimedfireh=zeros(N,M);
lhsf=zeros(N,M); lhsh=zeros(N,M);

% this is the rifle area fire term
areafiref=conv2(h,exp(-vf.*d*deltx*delty),'same');
% for an alternate artillery form of area fire, use:
% conv2(h,exp(-vf.*abs(d-optimalrange)*deltx*delty),'same')
areafiref=-gammaf1*f.*areafiref;
aimedfiref=-gammaf2.*f.*h;
lhsf=areafiref + aimedfiref;
lhsf=lhsf.*tau;

areafireh=conv2(f,exp(-vh.*d*deltx*delty),'same');
areafireh=-betah1*h.*areafireh;
aimedfireh=-betah2.*f.*h;
lhsh=areafireh+aimedfireh;
lhsh=lhsh.*tau;

f=f+lhsf;
h=h+lhsh;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%           Graphing Routine, 2D Snapshots                      %
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function subplotRB2D(f,zdata1,zdata2,currenttime)

set(f,'FontSize',10);
axis(f,[0 50 0 50]);
title(['Time = ' num2str(currenttime,'%6.5f')],'FontSize',10);
axis square
box('on');
hold('all');

contour1=contour(zdata1,'DisplayName','Force 1',...
    'LevelList',[1:1:10],'LineWidth',1);
contour2=contour(zdata2,'DisplayName','Force 2',...
    'LevelList',[1:1:10],'LineStyle',':','LineWidth',1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%           Graphing Routine, 3D Snapshots                        %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function subplotRB3D(f,zdata1,zdata2,currenttime)

colormap autumn

set(f,'FontSize',10);
axis(f,[0 50 0 50 0 14]);
title(['Time = ' num2str(currenttime,'%6.5f')],'FontSize',10);
view([25 31]);
grid('on');
hold('all');

%% Create surf
surf1=surf(zdata1,'EdgeColor',[1 0 0],'FaceAlpha',0.4,...
    'FaceColor',[1 0 0],'DisplayName','Force 1');
surf2=surf(zdata2,'EdgeColor',[0 0 1],'FaceAlpha',0.4,...
    'FaceColor',[0 0 1],'DisplayName','Force 2');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%           Graphing Routine, 2D and 3D for Movies                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function tripleplot(f,zdata1,zdata2,currenttime)

clf

set(f,'PaperPosition',[0.6345 6.345 20.3 15.23],...
'PaperSize',[20.98 29.68],'Position',[720 40 560 910]);
```

```
axes1=axes('Layer','top','Position',[0.1131 0.5209 ...
0.7974 0.4334],'CLim',[1 10],'FontSize',10,'Parent',f);
axis(axes1,[0 50 0 50]);
axis square
title(axes1,['Time = ' num2str(currenttime,'%6.5f')]);
box(axes1,'on');
hold(axes1,'all');

contour1=contour(zdata1,'Parent',axes1,'DisplayName','Force 1',...
    'LevelList',[1:1:10],'LineWidth',1);
contour2=contour(zdata2,'Parent',axes1,'DisplayName','Force 2',...
    'LevelList',[1:1:10],'LineWidth',1); %'LineStyle',':',

colorbar1=colorbar('peer',axes1,'EastOutside',...
'Box','on','FontSize',10);

axes2=axes('Position',[0.13 0.13 0.775 0.3363],...
'FontSize',10,'Parent',f);
axis(axes2,[0 50 0 50 0 12]);
zlabel(axes2,'Density');
view(axes2,[25 31]);
grid(axes2,'on');
hold(axes2,'all');

surf1=surf(zdata1,'EdgeColor',[1 0 0],'FaceAlpha',0.4,...
    'FaceColor',[1 0 0],'Parent',axes2);
surf2=surf(zdata2,'EdgeColor',[0 0 1],'FaceAlpha',0.4,...
    'FaceColor',[0 0 1],'Parent',axes2);

annotation1=annotation(...
   f,'textbox','Position',[0.05208 0.02183 0.8943 0.06],...
   'FitHeightToText','off','FontSize',10,...
   'String',{'Standard Simulation.','Comments Here.'});
```

Bibliography

[1] R. Weiner A. Gerisch, D. F. Griffiths and M. A. J. Chaplain. A positive splitting method for mixed hyperbolic-parabolic systems. *Numerical Methods for Partial Differential Equations*, 17:152–168, 2001.

[2] Boris Belousov. A periodic reaction and its mechanism. *Collection of short papers on radiation medicine for 1958*, 147:145, 1959.

[3] G. P. Boswell, H. Jacobs, F. A. Davidson, G. M. Gadd, and K. Ritz. A positive numerical scheme for a mixed-type partial differential equation model for fungal growth. *Applied Mathematics and Computation*, 138:321–340, 2003.

[4] T. Chapman, V. Mills, M. Kardos, C. Stothard, and D. Williams. The use of the Janus wargame simulation to investigate naturalistic decision-making: A preliminary investigation. Technical Report DSTO-TR-1372, Defence Science and Technology Institution, 2002.

[5] S. J. Deitchman. A Lanchester model of guerrilla warfare. *Institute for Defense Analyses*, pages 818–827, 1962.

[6] Andreas Deutsch and Sabin Dormann. *Cellular automaton Modeling of Biological Pattern Formation.* Birkhuser Boston, Cambridge, MA., 2005.

[7] J. M. Epstein. The adaptive dynamic model of combat. In *1992 Lectures in Complex Systems, V*, pages 437–459. Addison-Wesley, New York, 1993.

[8] J. M. Epstein. On the mathematical biology of arms races, wars and revolutions. In *1992 Lectures in Complex Systems, V*, pages 425–436. Addison-Wesley, New York, 1993.

[9] P. Flocchini, G. Guerts, A. Mignarelli, and N. Santoro. Convergence and aperiodicity in fuzzy cellular automata: revisiting rule 90. *Physica D*, 142:20–28, 2000.

[10] P. Gomez-Mourelo. From individual-based models to partial differential equations: An application to the upstream movement of elvers. *Ecological Modelling*, 188:93–111, 2005.

[11] David I. Graham and Rana A. Moyeed. How many particles for my Lagrangian simulations? *Powder Technology*, 125:179–186, 2002.

[12] D. Grunbaum and A. Okubo. *Modelling social animal aggregation.* Springer, New York, 1994.

[13] Daniel Grunbaum. Translating stochastic density-dependent individual behaviour with sensory constraints to an Eulerian model of animal swarming. *Journal of Mathematical Biology*, 33:139–161, 1994.

[14] D. S. Hartley and R. L. Helmbold. Validating Lanchester's square law and other attrition models. *Naval Research Logistics*, 42:609–633, 1995.

[15] Robert L. Helmbold. The constant fallacy: A persistent logical flaw in applications of Lanchester's equations. *European Journal of Operations Research*, 75:647–658, 1994.

[16] W. Hundsdorfer, B. Koren, M. van Loon, and J. G. Verwer. A positive finite difference advection scheme. *Journal of Computational Physics*, 117:35–46, 1995.

[17] A. Ilachinski. Land warfare and complexity, Part II: An assessment of the applicability of nonlinear dynamics and complex systems theory to the study of land warfare. Technical Report CRM 96-68, Center for Naval Analyses, 1996.

[18] A. Ilachinski. Irreducible Semi-Autonomous Adaptive Combat (ISAAC): An artificial life approach to land combat. Technical Report CRM 97-61.10, Center for Naval Analyses, 1997.

[19] A. Ilachinski. Exploring self-organized emergence in an agent-based synthetic warfare lab. *Kybernetes*, 32:38–76, 2003.

[20] N. K. Jaiswal and B. S. Nagabhushana. Combat modeling with spatial effects, reserve deployment and termination decision rules. *Computers and Operations Research*, 21:615–628, 1994.

[21] Ronald D. Fricker Jnr. Attrition models of the Ardennes campaign. *Naval Research Logistics*, 45:1–22, 1998.

[22] A. F. Karr. Lanchester attrition processes and theatre-level combat models. In *Mathematics of Conflict*, pages 89–126. Elsevier, North Holland, 1983.

[23] F.W. Lanchester. *Aircraft in Warfare.* Appleton, New York, 1916.

[24] M. K. Lauren. Characterising the difference between complex adaptive and conventional combat models. Technical Report 169, DOTSE Report, 1999.

[25] M. K. Lauren and R. T. Stephen. Map-Aware Non-uniform Automata (MANA) A New Zealand approach to scenario modelling. *Journal of Battlefield Technology*, 5:27–31, 2002.

[26] M. K. Lauren and R. T. Stephen. Using fractals to understand the role of entropy in complexity science: An examination of the MANA combat model. *Fractals (in preparation)*, 2002.

[27] C. T. Lee, M. F. Hoopes, J. Diehl, W. Gilliland, G. Huxel, E. V. Leaver, K. McCann, J. Umbanhowar, and A. Mogilner. Non-local concepts and models in biology. *Journal of Theoretical Biology*, 210:201–219, 2001.

[28] H. S. Lee, C. J. Matthews, R. D. Braddock, G. C. Sander, and F. Gandoler. A MATLAB method of lines template for transport equations. *Environmental Modelling and Software*, 19:603–614, 2004.

[29] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems.* Cambridge University Press, United Kingdom, 2002.

[30] A. J. Leverentz, C. M. Topaz, and A. J. Bernoff. Asymptotic dynamics of attractive-repulsive swarms, 2008.

[31] Ruey-Tarng Liu and Sy-Sang Liaw. Oscillatory Turing patterns in a simple reaction-diffusion system. *Journal of the Physical Korean Society*, 50:234–238, 2007.

[32] Frithjof Lutscher. Modeling alignment and movement of animals and cells. *Mathematical Biology*, 45:234–260, 2002.

[33] R.M.M. Mattheij, S.W. Rienstra, and J.H.M. ten Thije Boonkkamp. *Partial Differential Equations: Modeling, Analysis, Computation.* Society for Industrial & Applied Mathematics (SIAM), Philadelphia., 2005.

[34] Angelo Mignarelli. *The Dynamics of General Fuzzy Cellular Automata*, volume 3515/2005 of *Lecture Notes in Computer Science*, pages 351–359. Springer-Verlag, 2005.

[35] A. Mogilner and L. Edelstein-Keshet. A non-local model for a swarm. *Journal of Mathematical Biology*, 38:534–570, 1998.

[36] A. Mogilner, L. Edelstein-Keshet, L. Bent, and A. Spiros. Mutual interactions, potentials, and individual distance in social aggregation. *Journal of Mathematical Biology*, 47:353–389, 2003.

[37] K. W. Morton. *Numerical solution of convectiondiffusion problems.* Chapman & Hall, London, 1996.

[38] J. D. Murray. *Mathematical Biology.* Springer Verlag, New York, 1989.

[39] A. Okubo. *Diffusion and Ecological Problems.* Springer Verlag, New York, 1980.

[40] A. Okubo. Dynamical aspects of animal grouping: swarms, schools, flocks and herds. *Advances in Biophysics*, 22:1–94, 1986.

[41] G. M. Okubo. *Mathematical Models in Molecular ad Cellular Biology*, chapter 6.7, pages 523–567. Cambridge University Press, Cambridge, New York, 1980.

[42] Julia K. Parrish and William K. Hamner. *Animal groups in three dimensions.* Cambridge University Press, New York, 1997.

[43] N. Perry. Verification and validation of the fractal attrition equation. Technical Report DSTO-TR-1822, Defence Science and Technology Institution, 2006.

[44] V. Protopopescu, R. Santoro, R. Cox, and P. Rusu. Combat modeling with partial differential equations: The bidimensional case. Technical Report ONRL/TM-11343, Oak Ridge National Laboratory, 1990.

[45] V. Protopopescu, R. Santoro, and J. Dockery. Combat modeling with partial differential equations. *European Journal of Operations Research*, 38:178–183, 1989.

[46] V. Protopopescu, R. Santoro, J. Dockery, R. Cox, and J. Barnes. Combat modeling with partial differential equations. Technical Report ONRL/TM-10636, Oak Ridge National Laboratory, 1987.

[47] P. Rusu. Two dimensional combat modeling with partial differential equations. Technical Report ONRL/TM-10973, Oak Ridge National Laboratory, 1987.

[48] S. Sakai. A model for group structure and its behaviour. *Biophysics*, 13:82–90, 1973.

[49] W. E. Schiesser. *The Numerical Method of Lines: Integration of Partial Differential Equations.* Academic Press, San Diego, 1991.

[50] C. Spradlin and G. Spradlin. Lanchester's equations in three dimensions. *Computers and Mathematics with Applications*, 53:999–1011, 2007.

[51] R. Suzuki and S. Sakai. Movement of a group of animals. *Biophysics*, 13:281–282, 1973.

[52] Chad M. Topaz and Andrea L. Bertozzi. Swarming patterns in a two-dimensional kinematic model for biological groups. *SIAM Journal of Applied Mathematics*, 65:152–174, 2004.

[53] Alan Turing. The chemical basis of morphogenesis. *Phil. Trans. Of the Royal Society*, 237:5–72, 1952.

[54] R. Tyson, L. J. Stern, and R. J. LeVeque. Fractional step methods applied to a chemotaxis model. *Journal of Mathematical Biology*, 41:455–475, 2000.

[55] Y. Tyutyunov, I. Senina, and R. Arditi. Clustering due to acceleration in the response to population gradient: A simple self-organisation mode. *American Naturalist*, 164:722–735, 2004.

[56] Rune Vabo and Leif Nottestad. An individual based model of fish school reactions: predicting antipredator behaviour as observed in nature. *Fisheries Oceanography*, 6:155–171, 1997.

[57] Steve V. Viscido, Matthew Miller, and David S. Wethey. The response of a selfish herd to an attack from outside the group perimeter. *Journal of Theoretical Biology*, 208:315–328, 2001.

[58] Stephen Wolfram. *A New Kind of Science.* Wolfram Media, Inc, Champaign, Il., 2002.

List of Tables

List of Figures