

SIMULADOR DE CARDUMES EM AUTÔMATO CELULAR

Daniel Rezende Neves¹
Víctor Romário Paz de Jesus²
Christianne Orrico Dalforno³

Resumo

Algumas espécies de animais, tais como peixes, são capazes de se manterem em grupo sem a necessidade de um líder para organiza-los e guia-los (OBOSHI et al, 2003). O modelo biológico de cardume conceituado por Huth e Wissel (1994) propõe que um grupo é capaz de se manter unido através de simples interações com seus indivíduos vizinhos. Fundamentado neste modelo, este trabalho teve por objetivo o desenvolvimento de um simulador de cardume hábil suficiente para reproduzir o comportamento deste grupo de animais. Sua modelagem matemática foi embasada em autômatos celulares de forma a garantir que este simulador seja capaz de aproveitar as vantagens dos autômatos celulares em simular com grande qualidade os fenômenos naturais e artificiais (SOUSA, 2002). Contudo a utilização de modelos em autômatos celulares demanda uma grande carga de processamento, o que é um problema, principalmente para dispositivos acessíveis via web. Em vista disto, a computação paralela pode ser uma boa solução desde quando não exija um grande custo adicional como a montagem de um cluster, por exemplo. O desenvolvido deste simulador visa torna-lo um modelo de computação paralela para sistemas de deslocamento espaço-temporal. Portanto a escolha do simulador de cardume esta associada a sua fácil compreensão e desenvolvimento. Neste trabalho o interesse reside em utilizar o poder de processamento das placas de vídeo, que têm um baixo custo se comparadas a um cluster de computadores.

Palavras-chave: Simulador; Autômato celular; Processamento paralelo; CUDA; Java Script; GPU.

Abstract

Some species of animals, such as fish, are able to remain in the group without the need for a leader to organize and guide those (Oboshi et al, 2003). The biological model of fish schools conceptualized by Huth and Wissel (1994) proposes that a group is able to remain itself united by simple interactions with neighboring individuals. Based on this model, this research aimed to develop a simulator of fish schools to reproduce the behavior of this group of animals. Their mathematical model was based on cellular automata to ensure that this simulator is able to take advantage of cellular automata to simulate with high quality natural and artificial phenomena (SOUSA, 2002). However the use of cellular automata models demand a great amount of processing, which is a problem, especially for devices accessible by the web. Therefore, the parallel computing can be a good solution if it does not require a large additional cost like assembling a cluster, for example. The developed this simulator aims to make it a model of parallel computation for time-space displacement systems. Therefore the choice of have the simulator of fish schools done is associated with its easy understanding and development. In this research, the interest lies in using the processing power of video cards that have a low cost compared to a cluster of computers.

Keywords: Fish schooling simulator; Cellular automata; Parallel computation; CUDA; Java Script; GPU.

¹ Graduando em Engenharia Mecânica, Voluntário do Grupo de Pesquisa GANGES (Grupo de Aplicações e Análise Geoespaciais) - danielrezendeneves@gmail.com

² Graduando em Engenharia de Computação, Voluntário do Grupo de Pesquisa GANGES (Grupo de Aplicações e Análise Geoespaciais) – victor.romario.unifacs@gmail.com

³ MSC em Introdução a Pesquisa em Informática, Pesquisadora do Grupo GANGES (Grupo de Aplicações e Análise Geoespaciais) - christianne.dalforno@pro.unifacs.br

1. INTRODUÇÃO

Através de processos evolutivos alguns animais aprenderam a se organizar e se agrupar, de forma que pudessem aproveitar as vantagens da mutua ajuda para obter facilidade em achar comida e proteger-se de predadores (OBOSHI et al. 2003). Os cardumes, como título de exemplo e alvo do trabalho, são grupos onde os peixes permanecem em conjunto sem a necessidade de apresentar uma liderança, fato que chamou a atenção de muitos pesquisadores (OBOSHI et al, 2003), tais como Huth e Wissel.

O modelo analítico proposto por Huth e Wissel (1992) afirma que tais animais são capazes de se manter juntos por meio de simples interações extinguindo, dessa forma, a presença de líderes neste cardume. Neste modelo cada indivíduo pode exprimir uma dentre quatro reação de interação: repulsão, orientação paralela, atração e procura. Este modelo proposto veio com a intenção se mostrar capaz de simular um cardume, e de fato consegue. A comprovação deste modelo foi obtida através dos resultados do simulador desenvolvido por este grupo de pesquisadores.

O simulador proposto por este trabalho foi desenvolvido a partir de um modelo matemático em de autômatos celulares, onde a partir de simples códigos e uma regra universal foi modelado um universo bidimensional (matriz) capaz de simular um cardume através de partículas que se movem no espaço e no tempo ocupando alocações denominadas de células.

Há diversas maneiras de se projetar e executar uma simulação computacional. Neste projeto estamos especialmente interessados na abordagem baseada em sistemas de deslocamento de partículas utilizando um modelo matemático de autômatos celulares, o que demanda uma grande carga de processamento e pode ser um problema, principalmente para dispositivos acessíveis via web. Em vista disto desenvolvemos este simulador para ser acessível via web, em busca de alcançar uma ampla abordagem para este modelo. O grande desenvolvimento tecnológico das placas de vídeo torna possível hoje por em prática o conceito de computação paralela, que é ter o processamento de mais de uma linha de código ao mesmo tempo, de uma forma economicamente mais viável, pois não se faz necessário a utilização de cluster de computadores.

O grande objetivo deste trabalho é desenvolver um sistema em paralelo de deslocamento espaço-temporal de partículas para servir de modelo de paralelização para qualquer sistema afim. As placas que possuem o poder de fazer esta paralelização são chamadas de GPGPU (General Purpose Graphics Processing Unit - Unidade de Processamento Gráfico de Propósitos Gerais), as quais serão utilizadas neste simulador de

cardume. Nesta unidade de processamento, GPGPU, a paralelização é executada pela arquitetura CUDA.

2. METODOLOGIA

A versão atual 4.0 do simulador foi alcançada através de uma metodologia de pesquisas qualitativa, interpretando o sistema proposto por Huth e Wissel (1992 e 1994) e atribuindo variáveis. Portanto, não foram utilizados métodos estatísticos para o entendimento do fenômeno, uma vez que, o modelo no qual está embasado este simulador já está claramente descrito e fundamentado por Huth e Wissel (1992 e 1994).

3. AUTÔMATOS CELULARES

Os autômatos celulares foram conceituados formalmente por John Von Neumann em meados do século passado (século XIX) como um sistema de autorreprodução cujo objetivo é simular processos ou eventos naturais ou artificiais através de simples regras (SOUZA et al., 2004).

O simulador desenvolvido adota o modelo matemático dos autômatos celulares para reger as reações dos peixes. Os autômatos celulares são uma classe de modelos matemáticos que, devido as suas características, simulam com perfeição fenômenos naturais que ocorrem em diferentes escalas. O espaço de um autômato celular pode ser interpretado como um universo governado por uma regra onde muitos componentes simples ao atuarem juntos produzem complicados comportamentos (SOUZA, 2002). Isto permite estudar o cardume de forma mais detalhada estabelecendo uma relação entre o comportamento de seus indivíduos e seu comportamento macro.

Segundo Wolfram (1983), um autômato celular consiste estruturalmente num conjunto de células dispostas numa malha uniforme n-dimensional. Essas células podem assumir um dentre uma quantidade finita de estados. A evolução do autômato celular reside em definir o valor de suas células em espaços discretos de tempo. Cada célula é caracterizada por um estado definido por uma regra imutável a qual, geralmente, é definida pelo seu estado anterior e pelas células vizinhas. À vista disso, a evolução no tempo do sistema pode ser traduzida pela aplicação da regra sobre todas as células em paralelo.

Os componentes básicos de um autômato celular são (SOUZA et al., 2004):

- As células, elementos capazes de guardar um valor (estado);

- A rede ou universo n-dimensional no qual as células estão inseridas;
- A população. Esta consiste num conjunto de células que se distanciam em uma unidade, da qual se tem como referência e está sob análise, pra cada direção, podendo ser consideradas ou não as diagonais a depender do modelo de população adotado.
- A regra, consistindo num sistema lógico capaz de calcular novos valores de estados, com base nos obtidos na geração anterior, para um intervalo de tempo, de forma tal a definir uma nova geração.

Sendo assim este modelo matemático para simular os cardumes admite os peixes como células que mudam de estado ao interagir com células vizinhas almejando reproduzir o comportamento de um cardume.

Um dos desafios do uso de autômatos celulares é o grande volume de processamento (HAJNAL e BAJZÁT, 2011). Diversas são as implementações em paralelo deste tipo de modelo para resolver o problema de desempenho do sistema (SOUZA et al, 2004) (HAJNAL e BAJZÁT, 2011). Neste trabalho será desenvolvido um sistema de deslocamento espaço-temporal de partículas.

4. MODELO DE HUTH E WISSEL

Primeiramente é importante descrever de forma básica o modelo de Huth e Wissel (1994), no qual está baseado o simulador que este trabalho propõe.

No modelo de Huth e Wissel (1994) cada peixe presente no cardume muda sua direção e velocidade a cada intervalo de tempo seguindo uma regra universal. Nesta regra a velocidade e a direção dependem da direção, da velocidade e da posição relativa dos seus vizinhos. A influência de um vizinho pode ser expressa através de três reações diferentes: atração, orientação paralela, repulsão. Cada peixe é capaz de visualizar mais uma quantidade máxima de vizinhos. Para o modelo padrão são visualizados um máximo de quatro vizinhos que influenciam independentemente o seu alvo, portanto é necessário mesclar todas as influências existentes sobre o peixe.

O modelo de Huth e Wissel (1994) mescla as influências a partir de um método de média aritmética, definindo a velocidade escalar de um peixe através da média das velocidades de um grupo de vizinhos. Para definir as direções foi necessário associar direções a ângulos, sendo assim, estas são definidas através da média aritmética dos arcos de cada direção em potencial – para Huth e Wissel uma direção em potencial é a direção um vizinho

impõe sobre o peixe se apenas ele fosse considerado. Como dito anteriormente, o comportamento do cardume é expresso pelas reações de cada indivíduo deste ao interagir com os seus vizinhos. Sendo assim cada peixe pode expressar uma dentre as três reações. Tais reações são definidas pela distancia radial em que um vizinho se apresenta desde que este esteja dentro do campo de visão do peixe, o qual é dividido em três intervalos de distância denominadas áreas. Estas levam os nomes das reações que provocam (área de repulsão; área de orientação paralela; área de atração; área de procura ou área morta).

5. MECANISMO DE SIMULAÇÃO

O mecanismo de orientação dos cardumes de peixes do simulador proposto por este trabalho foi reproduzido por um autômato celular que simula as reações de cada peixe através de interações com os quais o rodeiam. O simulador desenvolvido, em Java Script, depende de alguns pressupostos fundamentais definidos por Huth e Wissel (1992) conforme citados abaixo:

- O universo de simulação é bidimensional;
- O tempo é quantizado e o movimento de cada indivíduo é redefinido a cada intervalo de tempo;
- Cada peixe nada de acordo com o mesmo modelo de comportamento. Isso garante que o modelo de cardume se mova sem um líder;
- O movimento do cardume não é afetado por interferências externas;
- Influências aleatórias são levadas em consideração para cada peixe, portanto a posição e a velocidade de cada indivíduo do cardume são construídas como variáveis estocásticas;
- O movimento de cada peixe é influenciado unicamente pelos vizinhos mais próximos, considerando que a visão e a linha lateral são os sentidos mais influentes num peixe;
- Buscar encontrar um modelo simples e possível. Somente um modelo simples promove a compreensão de seus resultados, em outras palavras, não há interesse em um modelo detalhado sobre o comportamento dos peixes, mas apenas os comportamentos decisivos para a organização de um cardume.

Assim como no modelo de Huth e Wissel (1994), neste trabalho não serão consideradas reações do cardume acompanhado de predadores, mas apenas as oriundas das interações dos

indivíduos do grupo entre si, pois o interesse não reside em reproduzir posicionamentos do grupo na presença de predadores, mas sim o comportamento deste ao se movimentar e se agrupar sem um líder.

O simulador correspondente a este projeto foi desenvolvido por etapas (inicialização; análise da matriz; análise dos vizinhos; novas coordenadas; nova matriz), onde cada uma destas corresponde a uma função fundamental do sistema, as quais estão ligadas entre si.

A primeira etapa, inicialização, é executada apenas uma vez no momento em que o programa é iniciado. Nesta etapa são criados o aquário e as vidas artificiais. O aquário representa a matriz principal do programa onde estarão dispostos os peixes. Dentro desta matriz bidimensional cinco números são possíveis de serem expressados, sendo estes os estados correspondentes do autômato celular. Cada estado expressa o objeto existente no aquário conforme a especificação abaixo:

- 0 = Água;
- 1 = Peixe na direção 1 (oeste);
- 2 = Peixe na direção 2 (norte);
- 3 = Peixe na direção 3 (leste);
- 4 = Peixe na direção 4 (sul);

Ao que diz respeito às vidas artificiais, cada uma delas é um vetor que guarda três variáveis: coordenada linha; coordenada coluna; estado. Ao passo que são criadas, estas são imediatamente armazenadas em uma matriz, denominada matriz dos peixes. A etapa de inicialização é representada pela função Início dentro do programa. Nesta função as variáveis dos peixes são determinadas randomicamente.

Na segunda etapa, nomeada análise da matriz, são percorridas todas as células da matriz principal à busca de um peixe. Cada vez que um peixe é encontrado a próxima etapa começa (análise dos vizinhos). Para definir quais células vizinhas serão analisadas é necessário compreender três conceitos: campo de visão total, campo de visão do peixe e direção.

O campo de visão total é definido pelo conjunto de quatro subcampos, um para cada direção primária. Conforme a ilustração abaixo o campo A é referente às três células à esquerda, campo B às três células acima, campo C às três células à direita, e por fim as três células abaixo representam o campo D. Por conseguinte, o campo de visão do peixe será a combinação de três subcampos mudando a depender da direção do peixe, pois o campo situado detrás do peixe faz parte da área morta, sendo assim este não será considerado em

análise. Como se pode entender para cada direção que um peixe pode assumir um diferente campo deixa de ser analisado considerando que o peixe somente apresentará quatro direções, as quais foram definidas anteriormente. Portanto o programa identifica a direção do peixe em questão e procura nos campos discretos a existência de um vizinho. Durante a execução desta função apenas o primeiro vizinho mais próximo de cada subcampo é considerado, pois mesmo que existam peixes, após este as reações que o peixe expressa sobre o vizinho estão hierarquizadas, tendo como preferência a reação sobre o vizinho de distância mais próxima.

			Campo B			
	Campo A		Peixe		Campo C	
			Campo D			

Figura 1 - Campo de visão total

Cada uma das três células presentes no subcampo representa uma área de reação tratada anteriormente. A primeira e mais próxima representa a área de repulsão, a segunda a área de orientação paralela e a última, por sua vez, representa a área de atração. Qualquer célula fora do campo de visão do peixe é considerada dentro da área morta ou de procura.

Com a finalidade de definir a reação resultante do peixe é necessário combinar as três reações expressadas sobre análise de cada subcampo. Sendo assim, para mescla-las adotamos um sistema de soma vetorial, conforme mostrado abaixo, onde cada subcampo apresenta um vetor capaz de armazenar duas variáveis (x , y), para as quais são adotados valores diferentes para cada reação. Na hipótese de um peixe não encontrar nenhum vizinho este permanecerá na procura e se movimentará na direção em que se apresenta.

$$somaX = A.x + B.x + C.x + D.x \quad (1)$$

$$somaY = A.y + B.y + C.y + D.y \quad (2)$$

Imediatamente após serem analisados os vizinhos e adquiridos todos os vetores, estes são somados na quarta etapa (novas coordenadas). Nesta serão redefinidas as coordenadas linha-coluna e os estados de cada peixe levando em consideração que os estados representam as direções. À medida que estas variáveis são redefinidas as mesmas são novamente armazenadas na matriz dos peixes substituindo os valores antigos pelos novos para cada

indivíduo do cardume. Por fim, após todos os peixes serem redefinidos a última etapa se inicia (nova matriz), nesta uma nova geração da matriz tomará o lugar da antiga a partir dos valores armazenados na matriz dos peixes e exibirá a nova matriz na tela. Ao final desta função a segunda etapa é novamente executada e se reinicia o ciclo.

O modelo atual do simulador, feito em Java Script, exibe na tela uma matriz expressa em *canvas*, onde foi definida a água pela cor azul e os peixes pela cor salmão, condizente com figura 2.

Foi desenvolvido um site por completo para inserir o simulador, utilizando as linguagens PHP, CSS, HTML e o JavaScript. Este site serve como suporte para o simulador sendo composto atualmente por uma *homepage*, uma página Atrigo, onde estarão publicadas todos os artigos referente a este trabalho, uma página referente ao simulador e uma página de contato, conforme a figura 2. O site desenvolvido é apenas um modelo experimental para testes, portanto ainda não foi lançado on line.

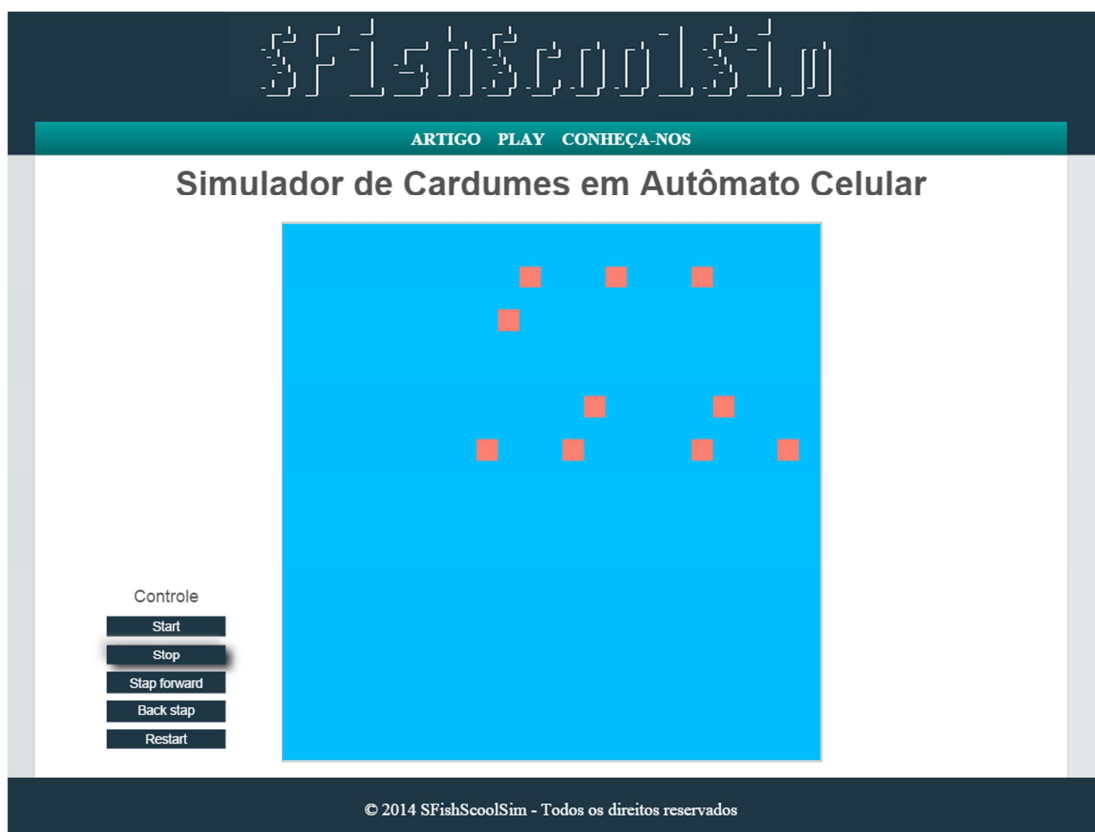


Figura 2 – Layout do Simulador

6. JAVA SCRIPT

Criada pela *Netscape* em 1995, o Java Script é uma linguagem de *script* que proporcionou dinamicidade às páginas web (FRANK et al.).

Apenas com o HTML e o CSS, só é possível criar uma página web estática. O HTML proporciona a estrutura e o CSS o estilo, mas é o Java Script que detecta qualquer coisa que ocorra numa página web, desde cliques em botões, redimensionamento de janelas ou dados em campo de texto (MORRISON, 2008).

Neste trabalho é proposta a integração da linguagem de programação JavaScripts com a plataforma CUDA (Compute Unified Device Architecture), que é uma extensão para linguagem C que possibilita a manipulação da GPU para realização de processamento de propósito geral e não apenas para processamento gráfico, com o objetivo de proporcionar melhor desempenho para as aplicações em Java Script, como o simulador deste trabalho.

7. CUDA CORE

O termo GPU é usado para o primeiro chip com funções integradas, capaz de realizar transformações, iluminação, recorte, ajuste de triângulos, texturização e renderização de imagens. (GULO, 2012) E este chip é utilizado pela GPGPU.

A única maneira para o programador interagir com a GPU, era por meio das bibliotecas OpenGL e DirectX (GULO, 2012).

Mais tarde, na GPU foi colocado um suporte para as transformações enviadas pela CPU, chamada de *vertex shaders* e depois foi complementada com o recurso de *fragment shader*, que permite criar efeitos e transformações diretamente nos pixels, antes de serem mostrados na tela do monitor. As operações passaram a ser realizadas através de Interface de Programação de Aplicações (APIs), conhecida como *shading language*. (GULO, 2012)

Mas este tipo de processador tinha limitações para o uso geral: era necessário saber a linguagem *shading* e computação gráfica, também era difícil a manipulação de dados em ponto flutuantes e se fazia necessário o monitoramento dos cálculos para evitar erros. (GULO, 2012)

A fim de resolver este problema, a NVIDIA desenvolveu a CUDA (Compute Unified Device Architecture) e construiu junto a ALU (Arithmetic and Logic Unit), para que a

unidade aritmética de ponto flutuante pudesse utilizar um conjunto de instruções para o uso geral e não somente processamento gráfico. (GULO, 2012)

Para aumentar o número de desenvolvedores da arquitetura CUDA, a NVIDIA incorporou à linguagem C, algumas *keywords* de funcionalidades próprias do CUDA, criando assim, um compilador CUDA C. (GULO, 2012)

Nesta extensão da linguagem C, existem três conceitos chave: uma hierarquia de grupos de threads; memória compartilhada; e barreiras de sincronização. (CARVALHO JUNIOR, 2008)

Os *threads* são utilizados pelos *kernels* para execução em paralelo. Estas *threads* são organizadas pelo programador em grandes blocos de threads, para que cooperem entre elas em uma barreira de sincronização e acesso compartilhado de memória (BUCK et al. 2008).

8. CONCLUSÃO

Os autômatos celulares permitem a simulação de sistemas emergentes, ou seja, aqueles cujo comportamento emerge da interação de seus componentes. Isto permite estudar o cardume de forma mais detalhada estabelecendo uma relação entre o comportamento de seus indivíduos e seu comportamento macro.

Para os testes do simulador definimos o aquário virtual com uma dimensão de 50x50 células e distribuimos 10 peixes aleatoriamente, porém a validação ficou comprometida por falta de dados reais para confrontar com os dados preditos pelo programa. Contudo longo da simulação pôde ser observada a formação de um ou mais grupos de peixes. Melhores resultados poderiam ser encontrados se fossem consideradas mais direções além das quatro direções primárias. Ainda assim o simulador cumpriu sua função de ser um sistema de deslocamento espaço-temporal de partículas.

O autômato foi desenvolvido na linguagem Java Script, para que pudesse ser executado na web e incluído ao site do projeto com sucesso. Como trabalhos futuros pretende-se desenvolver o simulador em paralelo no intuito de obter um melhor desempenho do sistema, utilizando a tecnologia CUDA core, e lançar o site no qual o simulador estará disponível.

REFERÊNCIAS

BUCK, I., NICKOLLS, J., GARLAND, M., SKADRON, K., NVIDIA. **Scalable Parallel Programming with CUDA**. 2008.

CARVALHO JUNIOR, Ademir José de Carvalho. **Uma Ferramenta MDA para a Plataforma CUDA**. 2008. 9p. Dissertação (Proposta de Trabalho de Graduação) - Universidade Federal de Pernambuco, Centro de Informática. 2008.

FRANK, D., SEIBT, L. **JavaScript**. 5p. Dissertação (Proposta de Trabalho de Graduação) – FIT – Faculdades de Informática de Taquara, Fundação Educacional Encosta Inferior do Nordeste.

GULO, Carloa Alex Sander Juvêncio. **Técnicas de paralelização em GPGPU aplicadas em algoritmo para remoção de ruído multiplicativo**. 2012. 80p. Dissertação (Mestrado) - Universidade Estadual Paulista, Instituto de Biociências, Letras e Ciências Exatas, São José do Rio Preto, 2012.

HAJNAL, Éva, BAJZÁT, Tamás. Ecological Modelling with Celular Automaton Software Implemented in GPU System. In: Óbuda University e-Bulletin. Vol. 2, No. 1, 2011.

HUTH, A. WISSEL, C. The Simulation of the Movement of Fish Schools. Journal of Theoretical Biology. Germany, n.156, p. 365-385, 1992.

HUTH, A. WISSEL, C. The Simulation of Fish Schools in Comparison with Experimental Data. Ecological Modelling, United States of America, n.75/76, p. 135-145, 1994.

MORRISON, Michael. **ALTA BOOKS. Use a Cabeça Java Script**. 2008

OBOSHI, Tamon KATO, Shohei MUTOH, Atsuko ITOH, Hidenori. A Simulation Study on the Form of Fish Schooling for Escape from Predator. Forma. Japan, n.18, p.119-131, 2003.

SOUSA, Sônia Alexandra F. S. Autômatos Celulares. 2002. 96f. Monografia (Graduação). Departamento de Ciência de Computadores. Faculdade de Ciências da Universidade do Porto – FCUP. Porto, Portugal, 2002.

SOUZA, Ariane Jeavaux, SILVA NETO, Antônio José, VASCONCELLOS, João Fávio Vieira, SANTOS, Luiz Orlando Emerick. Simulação do escoamento de fluidos incompressíveis com algoritmos de gás em rede e computação paralela. In: Proceedings of the 1th Brazilian Congress of Thermal Sciences and Engineering - ENCIT, 2004, Rio de Janeiro, Brasil.

WOLFRAM, Stephen. Statistical mechanics of cellular automata. **Reviews of Modern Physics**, vol. 55, No. 3, Julho 1983.