25. Meio Ambiente

Desenvolvimento de um modelo de inundação bidimensional acelerado por GPGPU

Carlotto, Tomas[1]; Innocente, Camyla[1]; Lee, Seungsoo[2]; Chaffe, Pedro[3]
thomas.carl@hotmail.com; camylainnocente@gmail.com; seungsoo_lee@apcc21.org; pedro.chaffe@ufsc.br
[1]Programa de Pós-Graduação em Engenharia Ambiental
[2]APEC Climate Center, Coreia do Sul
[3]Departamento de Engenharia Sanitária e Ambiental
[1,3]Universidade Federal de Santa Catarina, Brasil

Resumo

Inundações causam enormes prejuízos econômicos e afetam a vida de milhares de pessoas. Elaborar medidas para mitigar os efeitos das inundações é uma tarefa que exige o uso de modelos que simulem com precisão e rapidez o processo de inundação. Diante disso, os objetivos deste trabalho foram: (i) desenvolver uma implementação paralela de um modelo de inundação bidimensional para ser executado em unidades de processamento gráfico de propósito geral (GPGPU) e (ii) determinar o ganho de desempenho em comparação com uma versão sequencial equivalente. Como estudo de caso, fez-se a simulação da inundação do Campus Trindade da bacia hidrográfica da Universidade Federal de Santa Catarina. A versão paralela do modelo foi desenvolvida utilizando linguagem de programação CUDA C e uma estrutura baseada numa versão sequencial do modelo de inundação implementada em linguagem FORTRAN. Este modelo utiliza uma formulação 2D das equações de águas rasas discretizada pelo método de diferenças finitas. Para o desenvolvimento do código computacional utilizou-se o software Visual Studio Community 2013 e CUDA toolkit 8. As simulações foram realizadas em um computador equipado com processador Intel® Core™ i7-7700L 4.2GHz e GPU GeForce GTX 1060 6GB. Por meio das comparações entre os tempos de simulação verificamos que o modelo paralelo processado em GPGPU foi 70 vezes mais rápido que a versão sequencial executada na CPU, reduzindo o tempo de simulação de 12 horas para 10 minutos. Além disso, os resultados permitiram verificar a evolução do processo de inundação na bacia demonstrando que o uso de GPGPU é uma alternativa promissora na construção de modelos de inundação, para a previsão de cheias e emissão de alerta.

Palavras chave: modelo de inundação, GPGPU, CUDA.

Introdução

As ocorrências de inundações frequentemente causam grandes desastres gerando danos materiais e perdas de vidas (Blöschl et al., 2015 e Sætra, 2014). Os efeitos das inundações podem ser reduzidos pela definição de medidas de mitigação que variam desde a emissão de um alerta de perigo e o estabelecimento de um plano de evacuação até o desenvolvimento de estruturas de contenção (Horváth, 2014). Para orientar a elaboração do planejamento e a tomada de decisão de forma efetiva, torna-se importante a realização de previsões de ocorrências de inundações para diferentes cenários e condições meteorológicas (Sætra, 2014).

As inundações exibem diferentes comportamentos sendo que em casos mais críticos como rompimento de barragens e eventos extremos de vazão, se desenvolvem muito rapidamente exigindo ação imediata. Nestes casos as previsões de inundações demandam o uso de modelos computacionais que permitam fazer simulações rápidas e precisas (Horváth, 2014).

Os modelos baseados na solução das equações de águas rasas são usados há muitos anos para fazer previsões de possíveis cenários de inundação (Liang et al., 2009). Porém, o principal desafio na aplicação destes modelos é encontrar métodos numéricos e computacionais que proporcionem a redução do tempo de simulação (Brodtkorb et al., 2010).

Métodos numéricos de fácil implementação, baseados em esquemas de diferenças finitas tem sido usados com sucesso para aproximar a solução das equações de águas rasas em duas dimensões, toma-se como exemplo o método baseado em polígonos convexos com número arbitrário de lados proposto por Kawaike et al., (2000) que também foi usado em trabalhos como os de Thang et al., (2004) e Lee et al., (2014). Apesar de o método ser adequado, o principal problema é que a implementação computacional utilizada é baseada em estruturas de computação sequencial que costumam ser demoradas em simulações para grandes áreas e com eventos longos (Seitz, Kennedy, & Ransom, 2013).

Considerando que as simulações do processo de inundação geralmente requerem grandes domínios computacionais e são exigentes em termos de processamento, uma alternativa frequentemente utilizada no desenvolvimento de modelos tem sido a aplicação de técnicas de computação paralela (Seitz, Kennedy, & Ransom, 2013 e Liang et al., 2009).

As unidades de processamento gráfico de propósito geral (GPGPU) e a plataforma de computação paralela CUDA (*Compute Unified Device Architecture*) que utiliza linguagem de programação de alto nível,

semelhante a C e C++ tem surgido como uma alternativa de baixo custo para a realização de computação paralela oferecendo excelente custo benefício (Nickolls & Dally, 2010). A utilização de GPUs e algoritmos desenvolvidos para computação paralela proporcionam a solução eficiente das equações de águas rasas para a simulação de inundações em grandes áreas e tem sido uma das melhores opções para reduzir o tempo das simulações (Brodtkorb et al., 2010).

Considerando o grande potencial das GPUs e a facilidade da criação de códigos paralelos em CUDA, nós desenvolvemos uma implementação paralela de um modelo de inundação 2D utilizando linguagem de programação CUDA C com discretização espacial e temporal pelo método de diferenças finitas.

Aplicamos as versões paralela e sequencial do modelo de inundação 2D para simular um processo de inundação na bacia hidrográfica da Universidade Federal de Santa Catarina – Florianópolis (Figura 1). Por fim, realizamos a análise de desempenho comparando as versões sequencial e paralela do modelo de inundação 2D.

Objetivos

Desenvolver uma implementação paralela de um modelo de inundação bidimensional para ser executado em unidades de processamento gráfico de propósito geral (GPGPU)

Determinar o ganho de desempenho do modelo de inundação acelerado por GPGPU em comparação com um modelo sequencial processado em uma CPU convencional.

Materiais e Métodos

A Figura 1 mostra a bacia hidrográfica onde o Campus da Universidade Federal de Santa Catarina (UFSC) está localizado. A área de drenagem é de aproximadamente 4,09 km². As regiões baixas da bacia são urbanizadas, enquanto que as altas são compostas pelo Parque Municipal Maciço da Costeira e o Parque Municipal do Morro da Cruz.

Foram delimitadas oito sub-baciais de contribuição do Campus da UFSC (Figura 1). As áreas variam entre 1,19 e 0,07 km². Os principais canais de drenagem dentro do Campus são retificados, com paredes e fundo de concreto.

A precipitação e o nível da água no exutório da bacia foram medidos durante o ano de 2006. A discretização temporal dos dados é de um minuto. Os dados monitorados de dois eventos foram utilizados para calibrar o modelo hidrológico de chuva-vazão.

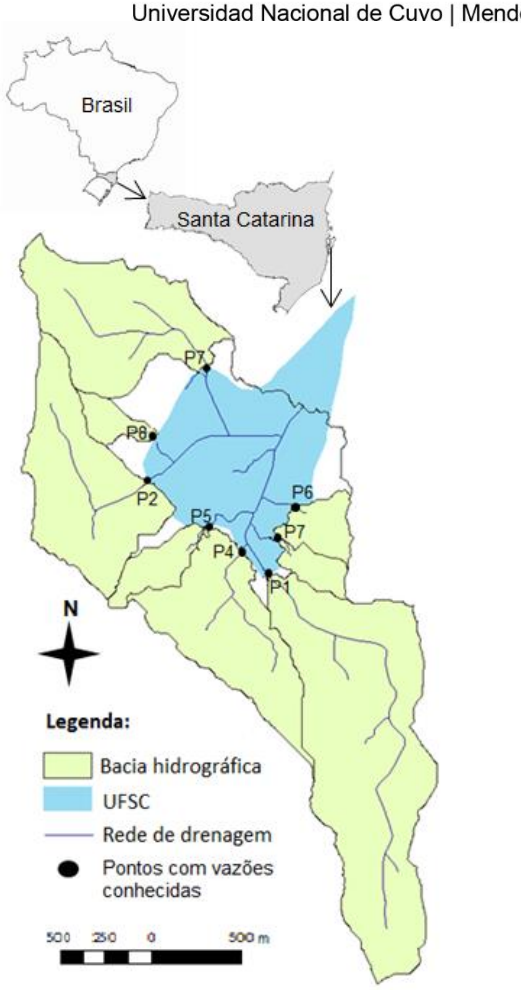

Figura 1. Área de estudo e localização dos pontos de vazões conhecidas.

Modelo hidrológico

O modelo hidrológico baseia-se na convolução do hidrograma unitário de acordo com a seguinte equação:

$$Q_t = \sum_{i=t-nc+1}^{t} P_i q_{t-i+1} \qquad (1)$$

em que, P é precipitação efetiva, q são as coordenadas do hidrograma unitário, Q são as coordenadas da vazão do escoamento superficial e nc é o número de coordenadas do hidrograma unitário.

Utilizou-se o modelo de hidrograma unitário de Mockus (1957). A porcentagem e a distribuição da chuva efetiva foram definidas pelo método do *Curve Number,* apresentado no livro de (Mockus et al., 1964).

Os parâmetros do modelo hidrológico foram calibrados e validados para a bacia da UFSC e regionalizados para as sub-bacias de contribuição. A função objetivo utilizada foi a diferença da altura de pico entre o hidrograma calibrado e o observado.

O hidrograma de projeto foi simulado para chuvas com tempo de retorno de 100 anos, com duração crítica de 180 minutos. A chuva foi calculada com base na equação de chuvas intensas de Florianópolis, dada por:

$$I_{máx} = \begin{cases} \dfrac{145\ {T_R}^{0,25}}{(t-1{,}18)^{0,34}} & t \leq 60 \\ \dfrac{597\ {T_R}^{0,32}}{(t+3)^{0,73}} & t > 60 \end{cases} \qquad (2)$$

em que T_R é tempo de retorno em anos, t é tempo em minutos e $I_{máx}$ é intensidade máxima em mm/h.

O escoamento de base foi considerado constante durante o evento. A vazão total foi o escoamento superficial somado ao escoamento de base.

A Figura 2 mostra o resultado da aplicação do modelo hidrológico nas bacias de contribuição e a chuva de projeto.

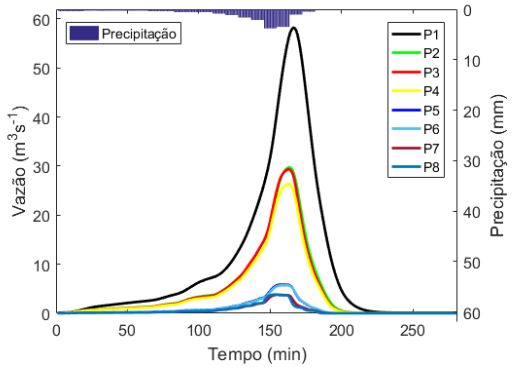

Figura 2. Precipitações e vazões utilizadas como entradas do modelo de inundação nos pontos P1, P2, P3, P4, P5, P6, P7 e P8.

Modelo de inundação

O modelo de inundação 2D utilizado neste trabalho é descrito pelas equações de águas rasas da seguinte forma:

- Equação da continuidade

$$\frac{\partial h}{\partial t} + \frac{\partial M}{\partial x} + \frac{\partial N}{\partial y} = r_e - q_{ex} \quad (3)$$

- Equações de momento

$$\frac{\partial M}{\partial t} + \frac{\partial (uM)}{\partial x} + \frac{\partial (vM)}{\partial y} = -gh\frac{\partial H}{\partial x} - f_1 \quad (4)$$

$$\frac{\partial N}{\partial t} + \frac{\partial (uN)}{\partial x} + \frac{\partial (vN)}{\partial y} = -gh\frac{\partial H}{\partial y} - f_2 \quad (5)$$

- Termos de fricção

$$f_1 = \frac{gn^2 M\sqrt{u^2+v^2}}{h^{4/3}} \quad (6)$$

$$f_2 = \frac{gn^2 N\sqrt{u^2+v^2}}{h^{4/3}} \quad (7)$$

em que h é a profundidade da água; $H = h + z$ sendo que z é a cota topográfica; u e v são as velocidades nas direções x e y respectivamente; M é o fluxo na direção x dado por uh; N é o fluxo na direção y dado por vh; r_e é a precipitação efetiva; q_{ex} é o termo que define a vazão de entrada na rede de drenagem (altura de água drenada por unidade de tempo em uma dada célula da rede de drenagem); t é o tempo; f_1 e f_2 são as componentes de fricção em que g é a aceleração da gravidade e n é o coeficiente de rugosidade de Manning.

A altura de água (h_d) que é drenada em uma dada seção do canal ao longo do tempo é definida pela seguinte equação:

$$\frac{\partial h_d}{\partial t} = \frac{Q_d}{A} = q_{ex} \quad (8)$$

em que: Q_d é a vazão acumulada na entrada do canal e A é a área da célula receptora.

Um esquema de discretização espacial e temporal das Equações 3, 4 e 5 pelo método de diferenças finitas utilizado por Kawaike et al., (2000), Thang et al., (2004) e Lee et al., (2014) é mostrado a seguir.

A Figura 3 apresenta uma célula elementar para exemplificar a discretização da equação da continuidade.

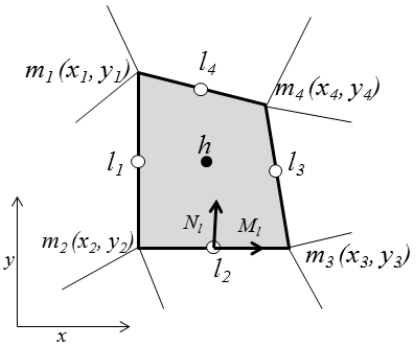

Figura 3. Célula ilustrativa para a discretização da equação da continuidade

Neste esquema de discretização o domínio computacional é dividido em polígonos convexos com número arbitrário de lados que são descritos em função dos valores das células adjacentes.

Com base na Figura 3 a formulação de diferenças finitas para a Equação 3 é escrita da seguinte forma:

$$\frac{h^{n+3}-h^{n+1}}{2\Delta t}+\frac{1}{A}\sum_{l=1}^{m}\left(M_l^{n+2}(\Delta y)_l - N_l^{n+2}(\Delta x)_l\right)=r_e-q_{ex} \quad (9)$$

em que: m e A são o número de lados e a área de uma dada célula elementar (Figura 3), respectivamente; M_l e N_l são as componentes do fluxo nas direções x e y para um lado l da célula; $(\Delta x)_{l_i}=(x_{i+1}-x_i)$ e $(\Delta x)_{l_i}=(x_{i+1}-x_i)$ sendo que $i=1,2,3,\dots m$ e n é o incremento no passo de tempo.

Para a discretização das equações de momento (Equações 4 e 5) considera-se a representação do volume de controle (VC) mostrado na Figura 4 (polígono em linha cheia) em que $(\widehat{M})_{out}$ e $(\widehat{N})_{out}$ são as componentes do fluxo nas direções x e y que saem do VC, respectivamente; $(\widehat{M})_{in}$ e $(\widehat{N})_{in}$ são as componentes do fluxo que entram no VC; d_1 e d_2 são as distâncias dos centroides das células 1 e 2 em relação ao lado L; $\tilde{h}_L$ é a altura da água interpolada para o lado L e d_L é a distância entre os centroides das células 1 e 2 adjacentes ao lado L.

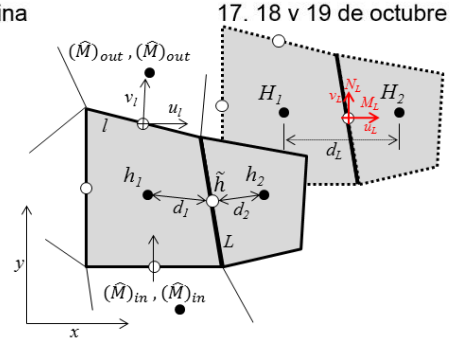

Figura 4. Representação de um volume de controle para discretização das equações de momento

De acordo com o método de diferenças finitas, as equações de momento tem a seguinte forma:

$$\frac{M_L^{n+2}+M_L^n}{2\Delta t}+M1+M2=-g\tilde{h}_L^{n+1}(\nabla H)_x+\cdots -gn^2\frac{\frac{M_L^{n+2}+M_L^n}{2}\sqrt{(u_L^n)^2+(v_L^n)^2}}{(\tilde{h}_L^{n+1})^{4/3}} \quad (10)$$

$$\frac{N_L^{n+2}-N_L^n}{2\Delta t}+N1+N2=-g\tilde{h}_L^{n+1}(\nabla H)_y+\cdots -gn^2\frac{\frac{N_L^{n+2}+N_L^n}{2}\sqrt{(u_L^n)^2+(v_L^n)^2}}{(\tilde{h}_L^{n+1})^{4/3}} \quad (11)$$

em que: M_1+M_2 e N_1+N_2 são os termos de convecção nas direções x e y, respectivamente, que podem ser escritos de acordo com as seguintes equações:

$$M1+M2=\frac{1}{A}\sum_{l=1}^{m}[u_l^n\widehat{M}_l^n(\Delta y)_l - v_l^n\widehat{M}_l^n(\Delta x)_l] \quad (12)$$

$$N1+N2=\frac{1}{A}\sum_{l=1}^{m}[u_l^n\widehat{N}_l^n(\Delta y)_l - v_l^n\widehat{N}_l^n(\Delta x)_l] \quad (13)$$

$(\nabla H)_x$ e $(\nabla H)_y$ são os gradientes de altura da superfície da água nas direções x e y

respectivamente, definidos com base na Figura 4 (polígono pontilhado) da seguinte forma:

$$(\nabla H)_x = \frac{H_2^{n+1} - H_1^{n+1}}{d_L} \cdot \frac{x_2 - x_1}{d_L} \quad (14)$$

$$(\nabla H)_y = \frac{H_2^{n+1} - H_1^{n+1}}{d_L} \cdot \frac{y_2 - y_1}{d_L} \quad (15)$$

Nos locais de fronteira solo/água onde pode haver a existência de fluxo utiliza-se a seguinte equação:

$$M(N) = \left(\frac{2}{3}\right)^{3/2} h\sqrt{gh} \quad (16)$$

em que h é a profundidade da água e g é a aceleração da gravidade.

No modelo apresentado neste trabalho utilizamos o esquema exposto acima com algumas simplificações assumindo a aplicação em malhas estruturadas em que o domínio computacional é composto por quadrículas regulares de mesma resolução espacial.

Implementação Paralela do Modelo

As GPGPUs apresentam grande poder computacional e tem sido cada vez mais utilizadas para resolver problemas gerais. Consequentemente, sistemas adaptados para a utilização de linguagens, bibliotecas e ferramentas (e.g. CUDA) para computação em GPUs tem evoluído bastante na última década, visto que estes dispositivos são encontrados na maioria dos desktops e laptops e são usados em supercomputadores para acelerar os cálculos em muitas aplicações do mundo real (Sætra, 2014).

Pensando em melhorar o desempenho computacional no processo de simulação de inundações, utilizamos computação paralela em GPGPU para implementar um modelo computacional de inundação 2D com linguagem de programação CUDA C/C++. Para isso, utilizamos o ambiente de programação do software Visual Studio Community 2013 com CUDA toolkit 8.0 disponibilizado gratuitamente pela empresa NVIDIA.

As principais etapas do modelo são apresentadas na Figura 5 em que as caixas em cor azul indicam os processos realizados na CPU e as caixas em cor verde indicam os processos realizados na GPU.As etapas de processamento apresentadas na Figura 5 foram divididas de acordo com a demanda computacional na realização dos cálculos. Para os processos computacionalmente mais exigentes foram desenvolvidas funções do tipo kernel, executadas na GPU com processamento paralelo. As outras etapas são executadas na CPU de forma sequencial. A versão sequencial do modelo de inundação 2D foi implementada em linguagem de programação FORTRAN sendo que o código original foi disponibilizado por Lee et al. (2014).

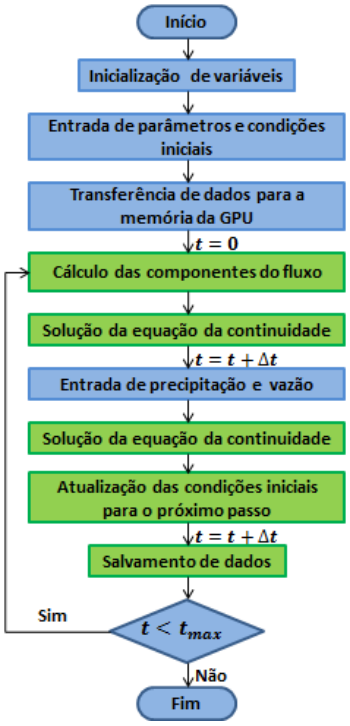

Figura 5. Principais etapas do modelo de inundação 2D em CUDA C/C++

Tanto o modelo sequencial quanto o modelo paralelo foram aplicados para simular o processo de inundação na área do campus da UFSC em Florianópolis (área em azul na Figura 1).

Nas simulações alimentamos a rede de drenagem em 8 pontos com vazões conhecidas e consideramos um evento de precipitação distribuída homogeneamente pela área da universidade. Os 8 pontos de entrada de escoamento são mostrados na Figura 1 (pontos pretos). As vazões em cada ponto e a precipitação distribuída no período de 4h e 41 minutos são apresentadas na Figura 2.

As simulações foram realizadas em um computador equipado com processador Intel® Core™ i7-7700L 4.2GHz e GPU GeForce GTX 1060 6GB. Utilizamos a ferramenta NVIDIA® Nsight™ Visual Studio Edition 5.4 para realizar a verificação da quantidade média de operações de ponto flutuante por segundo (FLOP/s) cada kernel do modelo paralelo conseguiu realizar durante a simulação, os resultados são apresentados na Figura 8. Essas estatísticas são utilizadas para avaliar como a demanda computacional é distribuída entre as diferentes etapas do código.

A análise de desempenho dos modelos foi realizada com base na comparação dos tempos de simulação de duas formas: (i) comparação dos tempos dos kernels processados na GPU com os tempos das funções equivalentes do modelo sequencial executado na CPU. (ii) comparação entre os tempos totais que cada modelo levou para completar a simulação.

Resultados e Discussão

As simulações do processo de inundação em duas dimensões realizadas com os modelos paralelo e sequencial mostraram a adequabilidade da formulação numérica e computacional para representarem o processo físico de inundação. Na Figura 6 são apresentados os resultados da simulação em GPU exibidas com intervalo de tempo de 30 minutos.

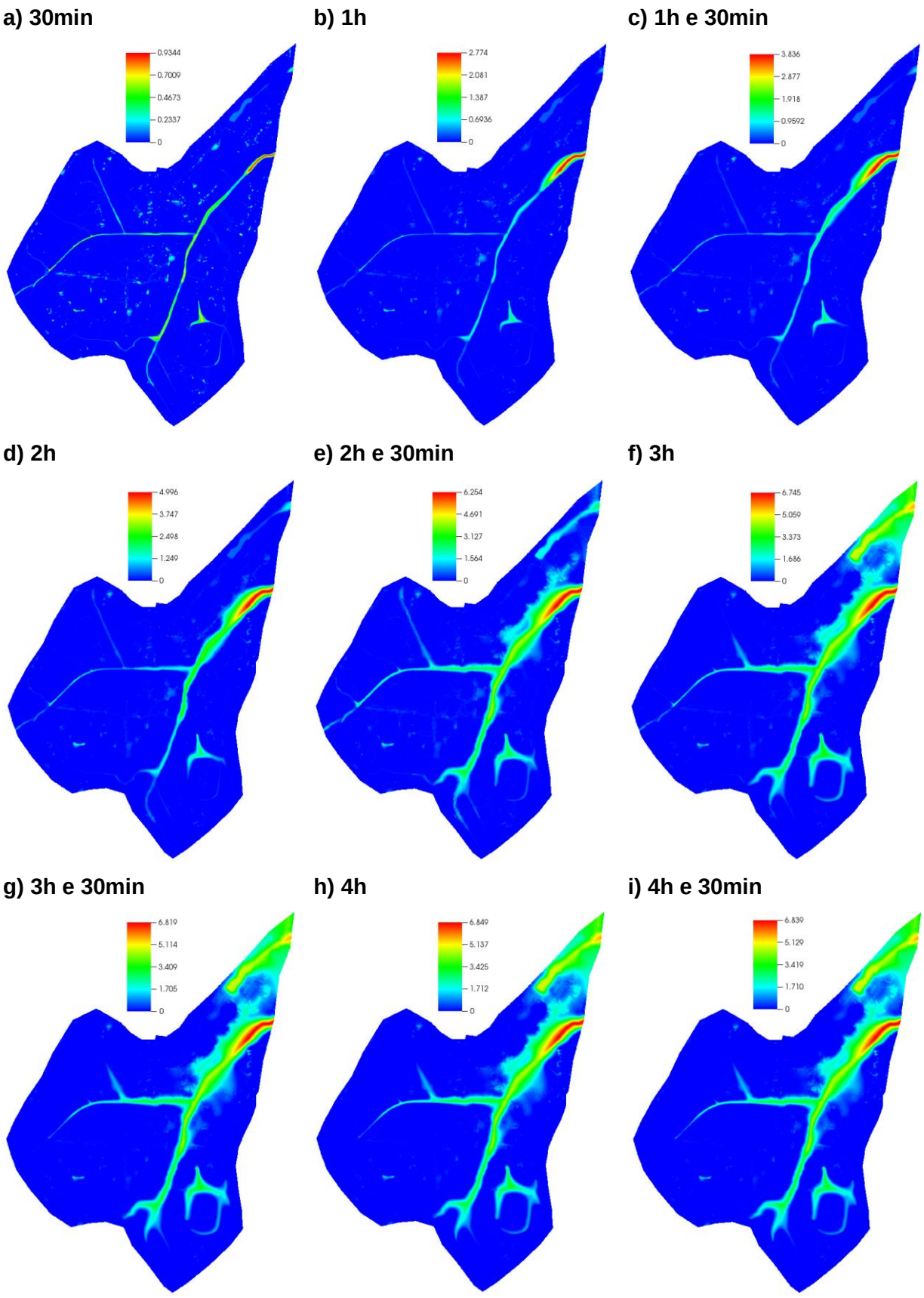

Figura 6. Resultados da simulação do processo de inundação ao longo do tempo mostrados com intervalo de tempo de 30 minutos.

Observa-se que a evolução da inundação ocorre com maior intensidade nas primeiras 2 horas e começa a estabilizar a partir de 3 horas simuladas, instante em que as entradas de vazão passam a ser muito pequenas.

Na Figura 7 mostra-se a variação do percentual de área inundada em função do tempo considerando três alturas diferentes do nível da água sobre a superfície do solo como estimadores da inundação. Os resultados reforçam o que já foi constatado visualmente na Figura 6 em que se verifica que o aumento da área inundada coincide com o período em que as vazões são maiores e sessa na ausência de vazão, deste modo os resultados representam adequadamente os hidrogramas apresentados na Figura 5.

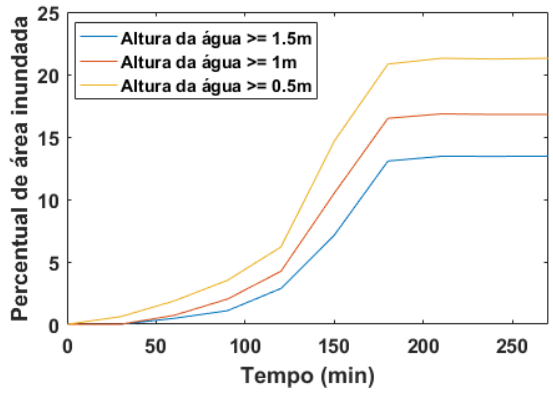

Figura 7. Percentual de área inundada em função do tempo.

Na Figura 8 mostra-se a quantidade de operações de ponto flutuante por segundo (FLOP/s) que ocorrem em cada kernel envolvido nos processos de aproximação da solução das equações de águas rasas na GPU. Verifica-se que a demanda computacional decresce da esquerda para a direita de modo que o kernel **flux** responsável pelo cálculo dos fluxos por meio da solução das equações de momento, realiza a maior quantidade de cálculos chegando a aproximadamente 120 bilhões de cálculos por segundo (Figura 8). Os demais kernels realizam etapas intermediárias de cálculo realizando menos FLOP/s.

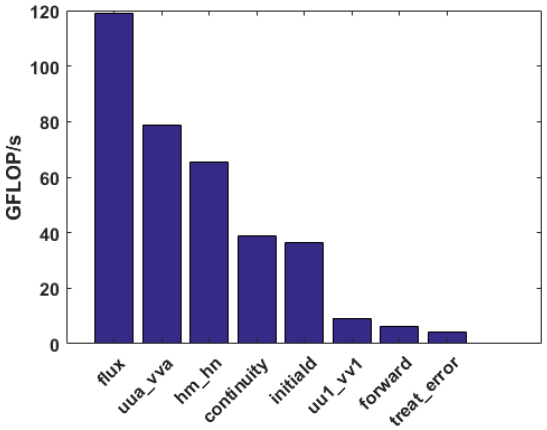

Figura 8. Número de operações de ponto flutuante por segundo para cada kernel processado na GPU

Na Figura 9 apresenta-se a comparação entre os tempos de processamento dos kernels que compõe o modelo de inundação processado na GPU e os tempos de processamento das funções que desempenham as mesmas tarefas no modelo sequencial em linguagem FORTRAN executado na CPU. O processamento em GPU é mais rápido em todas as etapas de execução do modelo em comparação com a versão sequencial.

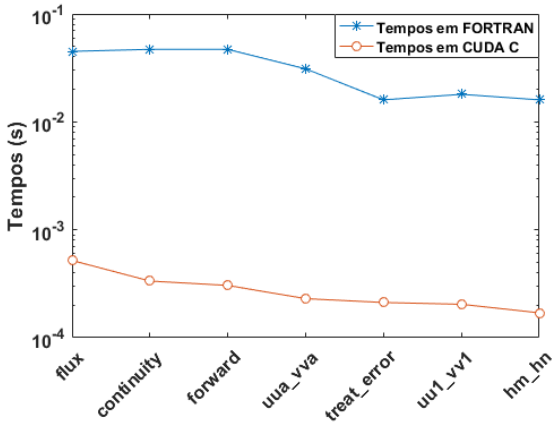

Figura 9. Tempos de execução dos principais kernels (CUDA) e funções (FORTRAN).

Os tempos de processamento de cada etapa dos modelos refletem no tempo total das simulações. O tempo total para o modelo paralelo processado na GPU realizar a simulação completa do processo de inundação foi 10min e 50s, já o modelo sequencial demorou 12h e 32min. Portanto, a implementação paralela proporcionou uma redução de aproximadamente 70 vezes no tempo de simulação mostrando que o uso de GPGPU é uma alternativa muito promissora para aplicações de modelagem de inundação em 2D.

Conclusões

Neste trabalho desenvolvemos uma implementação paralela de um modelo de inundação bidimensional acelerado por GPGPU. Conseguimos obter um bom desempenho computacional nas simulações do processo de inundação na área da Universidade Federal de Santa Catarina. A partir de comparações com um modelo sequencial que utiliza a mesma formulação numérica implementada em linguagem de programação FORTRAN mostramos que a versão paralela do modelo proporcionou uma redução de mais de 12 horas no tempo de simulação proporcionando um ganho de desempenho de aproximadamente 70 vezes. O modelo desenvolvido com o uso de unidades de processamento gráfico tem grande potencial para simulações rápidas que podem contribuir para a previsão de inundação e emissão de alertas de perigo.

Bibliografía

Blöschl, G., Gaál, L., Hall, J., Kiss, A., Komma, J., Nester, T., … Viglione, A. (2015). Increasing river floods: fiction or reality? *Wiley Interdisciplinary Reviews: Water*, *2*(4), 329–344. http://doi.org/10.1002/wat2.1079

Brodtkorb, A. R., Hagen, T. R., Lie, K. A., & Natvig, J. R. (2010). Simulation and visualization of the Saint-Venant system using GPUs. *Computing and Visualization in Science*, *13*(7), 341–353. http://doi.org/10.1007/s00791-010-0149-x

Horváth, Z. (2014). A two-dimensional numerical scheme of dry/wet fronts for the Saint-Venant system of shallow water equations. *International Journal for Numerical Methods in Fluids*, (October 2007), 601–629.

http://doi.org/10.1002/fld

Kawaike, K., Inoue, K., & Toda, K. (2000). Inundation flow modeling in urban area based on the unstructured meshes. *Hydraulic Engineering Software*.

Lee, S., Nakagawa, H., Kawaike, K., & Zhang, H. (2014). Urban Inundation Simulation Incorporating Sewerage System without Structure Effect. *Annuals of Disaster Prevention Res. Inst., Kyoto University., No. 57 B*, (57), 407–414.

Liang, W. Y., Hsieh, T. J., Satria, M. T., Chang, Y. L., Fang, J. P., Chen, C. C., & Han, C. C. (2009). A GPU-based simulation of tsunami propagation and inundation. *Lecture Notes in Computer Science*, *5574 LNCS*, 593–603. http://doi.org/10.1007/978-3-642-03095-6_56

Mockus, V. (1957). *Use of storm and watershed characteristics in synthetic hydrograph analysis and application*. Lathan.

Mockus, V., McKeever, V., & Rallison, R. (1964). *National Engineering Handbook*.

Nickolls, J., & Dally, W. J. (2010). The GPU computing era. *IEEE Micro*, *30*(2), 56–69. http://doi.org/10.1109/MM.2010.41

Sætra, M. L. (2014). *Shallow Water Simulations on Graphics Hardware*. University of Oslo.

Seitz, K., Kennedy, A., & Ransom, O. (2013). A GPU Implementation for Two-Dimensional Shallow Water Modeling. *arXiv Preprint*, 9. Retrieved from http://arxiv.org/abs/1309.1230

Thang, N. T., Inoue, K., Toda, K., & Kawaike, K. (2004). A model for flood inundation analysis in urban area: verification and application. *Annuals of Disaster Prevention Res. Inst., Kyoto University., No. 47 B*, (47).

Agradecimentos

Agradecemos a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pela bolsa de doutorado do primeiro autor.