



16, 17 e 18 de setembro de 2014  
Hotel Maksoud Plaza  
São Paulo – SP

## **Efeito dos erros de truncamentos e arredondamento em modelos chuva-vazão aplicados à drenagem na fonte**

### **Rounding and truncation errors effects in rainfall-runoff models applied at source control scale**

**Vitor Gustavo Geller<sup>1</sup>; Raviel E. Basso<sup>2</sup>; Daniel G. Allasia<sup>2</sup>; Rutinéia Tassi<sup>2</sup>**

1 Universidade Federal de Santa Maria, vitorgg\_hz@hotmail.com; 2 Universidade Federal de Santa Maria.

*Palavras-Chave: Hydrolib, IPHS1, comparação.*

*Key Words: Hydrolib, IPHS1, comparison.*

## **1. INTRODUÇÃO**

Ferramentas computacionais para modelagem hidrológica e hidráulica vêm sendo desenvolvidas há algumas décadas, sendo capazes de executar cálculos de forma rápida e precisa. O aprimoramento destes modelos deve acompanhar, por sua vez, as melhorias no hardware disponível. Na área hidrológica, alguns modelos já são consagrados por apresentarem bons resultados em determinadas situações. Um bom exemplo é o modelo hidrológico IPHS1, criado na década de 80 e aperfeiçoado nos anos seguintes (Ex, VIEGAS FILHO *et al*, 2004; TASSI *et al*, 2004). O IPHS1 apresenta ótimos resultados chuva-vazão para médias e grandes bacias. Contudo, ele possui limitações na representação de unidades menores de simulação, a exemplo dos pequenos planos de escoamento em áreas urbanas, em função de simplificações na sua formulação e escolha da precisão do seu código.

O IPHS1 foi programado em Fortran, com uma interface desenvolvida em Delphi (VIEGAS FILHO *et al.*, 2004), utilizando para as operações de ponto flutuante variáveis do tipo REAL de 4 bits. Em geral, este tipo de número se comporta de maneira semelhante aos números reais, com uma precisão de 6 casas decimais (PEREIRA CRISTO, 2003); no entanto, há muitas inconsistências entre o comportamento dos números de ponto flutuante em base 2 (binário) e os números reais, mesmo em casos muito simples. Entre as principais inconsistências que provocam erros no cálculo de ponto flutuante, podem ser citadas: arredondamento, arredondamento das operações aritméticas (por exemplo,  $2/3 = 0,666667$ ), cancelamento (por exemplo, a subtração de dois números muito próximos), erros de transbordamento (*underflow ou overflow*), ou seja, trabalhar com valores acima ou abaixo dos permitidos pela precisão, operação impossível (por exemplo, raiz quadrada de um número com resultados imaginários).

Para contornar estas limitações, linguagens clássicas como o Fortran implementaram números de dupla precisão (REAL\*8 ou simplesmente DOUBLE), com impacto significativo na velocidade



16, 17 e 18 de setembro de 2014  
Hotel Maksoud Plaza  
São Paulo – SP

de processamento. Já linguagens mais atuais, a exemplo do PYTHON, utilizam simplesmente números com 53 bits de precisão. A linguagem gratuita PYTHON foi criada no início da década de 90, por Guido van Rossum no Instituto Nacional de Matemática da Computação da Holanda (CWI), cujo objetivo principal era auxiliar engenheiros e físicos no processo de programação (COELHO, 2007). A sua extrema precisão deriva do aperfeiçoamento dos componentes de hardware dos computadores, a partir da década de 1990, eliminando muitas limitações de gerações anteriores, a exemplo do grande tempo computacional requerido para a simulação de alguns processos.

Além de o PYTHON ser gratuito, ele é uma linguagem de programação que pode ser orientado a objetos, possuiu sintaxe simples e ampla documentação, tem várias facilidades gráficas, pode ser utilizado em ambiente multiplataforma (Windows, Linux, Android, Ios, etc), possuindo grande quantidade de bibliotecas disponíveis (LUTZ, 2009). Soma-se a isso o fato de esta ser a linguagem de programação escolhida pelos softwares de geoprocessamento, como QGIS e ArcGIS. Assim, considerando as vantagens do PYTHON foi desenvolvida uma biblioteca denominada de Hydrolib (*Hydrologic Library* – Allasia *et al.*, 2014), que permite implementar facilmente modelos hidrológicos chuva-vazão, combinando diferentes metodologias, incluindo desde a geração de eventos de projeto a partir de equações IDF até a geração de escoamento com modelos consagrados na literatura. Essa biblioteca possibilita reduzir o trabalho e tempo gasto pelo usuário na entrada de dados para o cálculo de operações hidrológicas diversas.

Este artigo apresenta os resultados obtidos a partir do uso da biblioteca Hydrolib, buscando destacar os problemas relacionados aos erros de truncamento e arredondamento quando utilizados softwares não preparados para a representação de processos hidrológicos em pequenas bacias hidrográficas, a exemplo do IPHS1.

## 2. MATERIAL E MÉTODOS

Para a comparação do resultado em diferentes escalas de análise foram representadas nos Hydrolib e IPHS1 seis bacias hipotéticas, com áreas de drenagem que variaram entre 10 km<sup>2</sup> e 0,0001 km<sup>2</sup> (10 km<sup>2</sup>, 1 km<sup>2</sup>, 0,1 km<sup>2</sup>, 0,01 km<sup>2</sup>, 0,001 km<sup>2</sup> e 0,0001 km<sup>2</sup>), sendo que aquelas de pequena área representativas dos planos de escoamento em áreas urbanas e que normalmente representam a escala de planejamento para o controle na fonte.

Para a simulação foi utilizado um hietograma de projeto obtido a partir da equação IDF do posto IPH da cidade de Porto Alegre para uma duração de 1440 minutos e tempo de retorno de 10 anos. A distribuição temporal do evento chuvoso, discretizado a cada 300 segundos, foi realizada com a metodologia dos blocos alternados. Para a determinação da chuva efetiva foi utilizado o método do CN (Curve Number) do SCS, com valor igual a 85 (bacia urbana – bastante urbanizada), e a propagação do escoamento superficial foi feita com o método do Hidrograma Unitário Triangular do SCS. Ao final da simulação com os dois modelos, os hidrogramas gerados foram analisados para verificar as possíveis divergências nos resultados obtidos.



### 3. RESULTADOS E DISCUSSÃO

Na Figura 1 são apresentados os resultados da modelagem realizada. Observa-se no resultado a semelhança nos hidrogramas para as bacias de maiores (Figuras 1(a) a 1(c)), com pequenas diferenças temporais, em função de arredondamento/truncamento do  $\Delta t$  em torno do qual se distribuem os blocos de chuva na metodologia dos blocos alternados. Além disso, verifica-se e uma deficiência na representação do hidrograma com o IPHS1, especialmente na Figura 1(c), onde se observa um hidrograma “serrilhado” no período de ascensão e recessão.

Para bacias de menor área (Figuras 1(d) a 1(f)), a divergência de resultados entre o Hydrolib e IPHS1 é acentuada. Para a menor bacia simulada (Figura 1(f)) o IPHS1 não foi capaz de gerar o escoamento, enquanto para as duas outras bacias a geração do hidrograma foi comprometida, em razão da baixa precisão utilizada na programação do modelo. Os resultados com o Hydrolib permitem concluir que a precisão utilizada no PYTHON foi determinante para uma melhor representação do processo simulado para essas bacias. O mesmo tipo de problema de precisão foi identificado para outros softwares amplamente utilizados em drenagem urbana, a exemplo do Storm Water Management Model (SWMM), não discutido neste artigo em razão da limitação de espaço.

### 4. CONCLUSÃO

A partir dos hidrogramas gerados por pelo IPHS1 e Hydrolib, pôde-se observar que o modelo Hydrolib apresenta resultados muito similares aos do IPHS1 para bacias de grandes áreas; já para bacias de pequenas áreas observa-se grande divergência de resultados. A diferença citada ocorre, pois os valores de vazão gerados nas bacias de pequena área (drenagem na fonte) são da ordem de 0,001 m<sup>3</sup>/s, e como Fortran (usado no IPHS1) trabalha com apenas 4 bits de números binários para aproximação decimal, a precisão dos resultados é insuficiente para a grandeza dos valores do hidrograma de projeto, assim o resultado gerado é pobre em informações. A precisão do PYTHON no modelo Hydrolib permitiu uma melhor simulação chuva-vazão para as pequenas áreas de drenagem. Diferenças similares também foram observadas quando incluída na análise o Modelo SWMM, embora não foi possível determinar, neste caso, se foi problema numérico do programa ou da própria interface.

Os resultados encontrados neste estudo destacam a necessidade de cuidados na utilização das ferramentas computacionais existentes, principalmente quando aplicadas a estudos voltados para a drenagem na fonte, caracterizada por pequenas áreas de drenagem. Evidencia-se, também, a necessidade de desenvolvimento de ferramentas computacionais adequadas para esta situação, a exemplo da Hydrolib que foi testada.

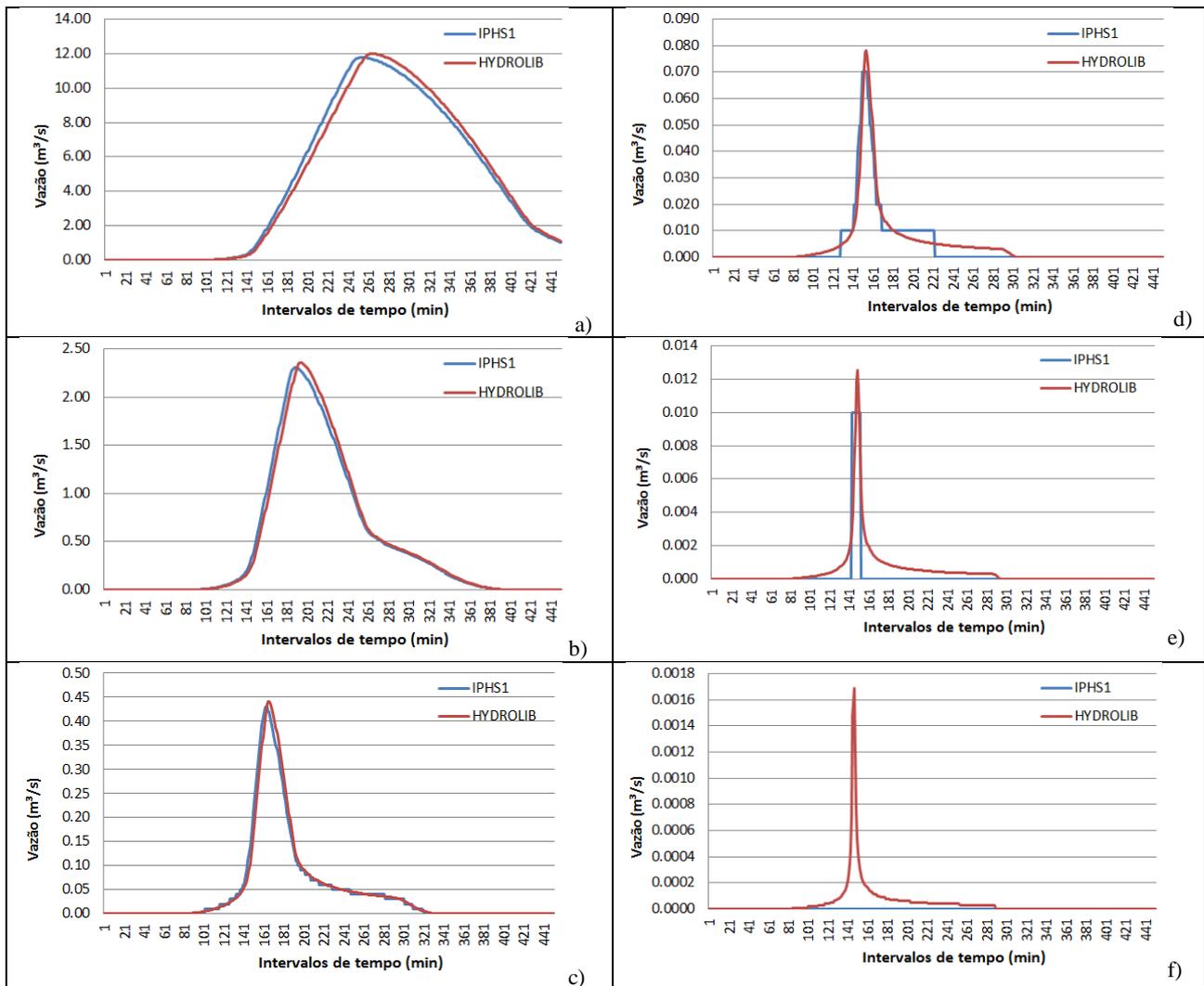


Figura 1: Comparações de hidrograma de projeto para ambos softwares com diferentes áreas: a) 10 km<sup>2</sup>; b) 1 km<sup>2</sup>; c) 0,1 km<sup>2</sup>; d) 0,01 km<sup>2</sup>; e) 0,001 km<sup>2</sup> f) 0,0001 km<sup>2</sup>.

## 5. REFERÊNCIAS

- ALLASIA, D. GELLER, V. TASSINARI, L. TASSI, R. Biblioteca para modelagens hidrológicas Hydrolib. Manual de usuário e fundamentos (em preparação). Santa Maria, UFSM. 2014.
- COELHO, F. C., Computação Científica com Python: Uma introdução para programação para cientistas. Petrópolis – RJ. 1 ed., 269 p., 2007.
- LUTZ, M.; ASCHER, D., Aprendendo Python. Porto Alegre – RS. 2 ed., 568 p., 2007.
- VIEGAS FILHO, C. E. M, TUCCI, VILLANUEVA, A., D. G. ALASIA P., TASSI, T., & R. C.F. AMÉ. Manual do Usuário IPHS1 para Windows, v. 2.11 – Versão PDF. IPH/UFRGS, Porto Alegre RS, 46 p. 2004.
- TASSI, R, ALLASIA, D. VIEGAS FILHO, C. E. M, TUCCI, VILLANUEVA, A., & R. C.F. DAMÉ. Manual de Fundamentos do IPHS1 – Versão Preliminar – em PDF. IPH/UFRGS, Porto Alegre RS, 45 p. 2004
- PEREIRA CRISTO, H. Programação em Linguagem FORTRAN. Disponível em: <http://www.inf.ufes.br/~thomas/fortran/tutorials/helder/fortran.pdf>. Belo Horizonte. Junho/2003