

DESENVOLVIMENTO DO MODELO HIDROLÓGICO LASH EM MATLAB®

MARCELLE MARTINS VARGAS¹; MAÍRA MARTIM DE MOURA²; CARLOS ROGÉRIO DE MELLO³; ZANDRA ALMEIDA DA CUNHA⁴; TAMARA LEITZKE CALDEIRA BESKOW⁵; SAMUEL BESKOW⁶

¹Discente do PPG Recursos Hídricos / CDTec / UFPel – marcellevarg@gmail.com;

²Discente do PPG Recursos Hídricos / CDTec / UFPel – martimdemoura@gmail.com;

³Docente do Departamento de Recursos Hídricos / UFLA – crmello@ufla.br;

⁴Discente do PPG Recursos Hídricos / CDTec / UFPel – zandraacunha@gmail.com;

⁵Docente do CEng / UFPel – tamaraleitzkecaldeira@gmail.com;

⁶Docente do CDTec / UFPel – samuel.beskow@ufpel.edu.br.

1. INTRODUÇÃO

Nas últimas décadas, a hidrologia tem testemunhado grandes avanços, especialmente devido à ampliação de bases de dados e ao desenvolvimento de recursos computacionais (SLATER et al., 2019). Nesse contexto, a construção de modelos hidrológicos em linguagens de programação interpretadas ganha espaço e margeia a utilização de artifícios robustos.

Dentre os modelos hidrológicos existentes, o modelo brasileiro *Lavras Simulation of Hydrology* (LASH) tem se destacado por sua crescente aplicabilidade no Brasil, já tendo sido avaliado em bacias hidrográficas de diferentes biomas (<https://rpubs.com/gphidrologia/baciaslash>). As principais vantagens do LASH, frente a outros modelos, é a quantidade reduzida de informações de entrada necessárias e de um número pequeno de parâmetros a serem calibrados. O histórico de desenvolvimento do LASH foi pautado em três versões, a saber: i) a 1ª versão, desenvolvida em planilhas eletrônicas sob forma semi-distribuída por sub-bacias (MELLO et al., 2008); ii) a 2ª versão, desenvolvida em ambiente de programação Delphi, além da implementação da discretização distribuída por células e o módulo de calibração automática (BESKOW et al., 2011); e iii) a 3ª versão (em Delphi) que contou com o desenvolvimento de dois módulos para tratamento dos dados de entrada ao LASH e, ainda, a implementação da discretização espacial semi-distribuída por sub-bacias (CALDEIRA et al., 2019).

O sucesso de aplicação das versões anteriores do LASH em bacias hidrográficas distintas remete ao seguinte questionamento: por que não continuar utilizando o LASH nas versões em planilhas eletrônicas ou Delphi? Apesar dos bons resultados, é necessário que a estrutura computacional do modelo acompanhe as ferramentas disponíveis, permitindo um *upgrade* de programação e de funcionalidades do modelo. Tal *upgrade* permite, por exemplo, acoplar novos módulos (e.g. propagação de vazões) e esquemas de discretização espacial, testar novas funcionalidades (e.g. algoritmos multiobjetivos) e melhorar o desempenho de execução do modelo. Estas premissas seriam bastante difíceis ou inviáveis de serem implementadas em planilhas eletrônicas ou Delphi, devido às limitações ou dificuldades impostas por esses ambientes. Cabe ressaltar que, na versão atual do modelo, estão implementados o método de propagação de Muskingum-Cunge e o algoritmo de calibração mono-objetivo *Shuffled Complex Evolution* (SCE-UA) (DUAN; SOROOSHIAN; GUPTA, 1992).

Frente ao exposto, este trabalho objetiva introduzir a implementação do modelo hidrológico LASH em ambiente MATLAB®, associado com as melhorias acerca: i) do módulo de propagação de vazões em rios; ii) do acoplamento de um algoritmo multiobjetivo; e iii) da otimização do tempo de processamento do modelo.

2. METODOLOGIA

A implementação do LASH em MATLAB® foi estruturada tendo como unidade de análise a sub-bacia hidrográfica e o passo de simulação diário (Figura 1), seguindo os baixos custos operacionais (CALDEIRA et al., 2019). Todavia, a sintaxe de programação do MATLAB® e a forma como o modelo está estruturado permitem, facilmente, que a calibração seja executada com o modelo de forma concentrada.

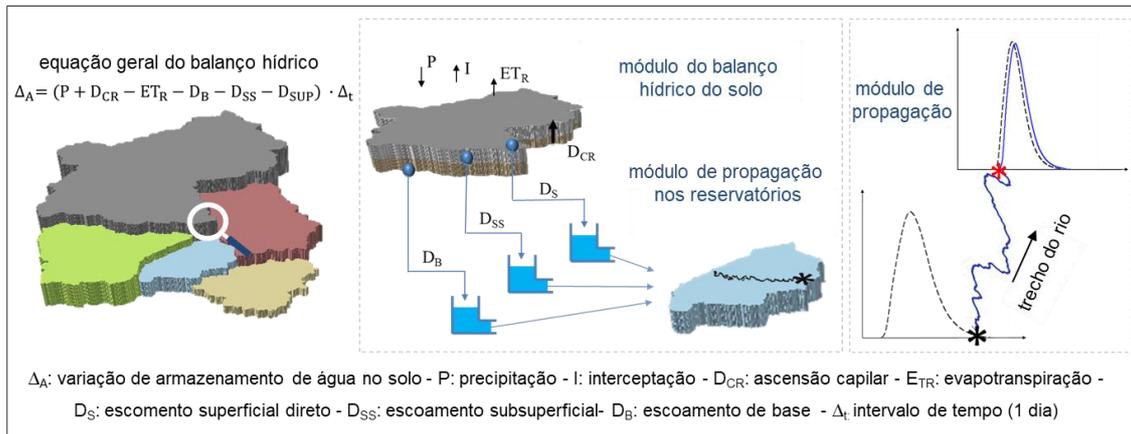


Figura 1 – Esquemática dos módulos do modelo hidrológico LASH.

Em relação ao método de propagação, foi adaptada e acoplada ao LASH a função não linear da Onda Cinemática, resolvida de forma implícita pelo método de Newton-Raphson, desenvolvida por Paiva (2007) em MATLAB®. Os dados necessários à função são: o hidrograma a ser propagado, o intervalo de tempo, o número de subtrechos em que o curso d'água será dividido, o coeficiente de rugosidade de Manning, o comprimento, a largura e a declividade de cada trecho de propagação. O método da Onda Cinemática é computado por trechos de propagação, logo, apenas bacias hidrográficas constituídas por sub-bacias com tempo de concentração superior a 1 dia são condicionadas ao método.

O *Multi-Algorithm Genetically Adaptive Multiobjective* (AMALGAM) foi acoplado ao LASH utilizando o *script* implementado em MATLAB® por Vrugt e Robinson (2007). Segundo os autores, o AMALGAM combina dois conceitos: a pesquisa multimétodo simultânea e a criação de filhos adaptativa para garantir uma solução rápida e computacionalmente eficiente para problemas de calibração multiobjetivo.

Por fim, outra necessidade iminente ao LASH foi a melhoria do tempo de processamento para a calibração do modelo, que nas versões anteriores era de 48h ~ 72h, dependendo do número de sub-bacias e da extensão das séries hidrológicas. Portanto, foi utilizado o artifício de globalização das variáveis, que equivale a um *pointer* na memória principal, ou seja, como as variáveis globais são acessíveis em qualquer seção do MATLAB®, não há necessidade de repassá-las às funções, otimizando o tempo de processamento.

Com vistas a verificar os objetivos delineados neste estudo acerca do desenvolvimento do LASH em MATLAB®, foram utilizados como estudos de caso: i) a bacia hidrográfica do rio Xingu (BHRX), a qual possui área de drenagem de 442.000 km², especialmente para analisar a melhoria da propagação; e ii) a bacia hidrográfica do rio Camaquã (BHRC), a qual possui área de drenagem de 15.571 km² a fim de avaliar a viabilidade do AMALGAM.

3. RESULTADOS E DISCUSSÃO

O LASH foi estruturado por meio de dois *scripts* *.m: i) LASH.m, o qual remete à importação da base de dados computada no ArcLASH (SANTOS et al., 2019), alocação em vetores específicos, definição das constantes iniciais do modelo, globalização das variáveis indexadas e definição da função de calibração; e ii) LASH_func.m, o qual concerne em uma função contendo a estrutura de repetição, por sub-bacia e por intervalo de tempo dos componentes do ciclo hidrológico. O LASH.m foi desenvolvido em 362 linhas e o LASH_func.m, em 740 linhas, totalizando 1.102 linhas de código considerando todas as funções implementadas. Em um comparativo com a 2ª e a 3ª versão (em Delphi), por exemplo, houve uma redução expressiva no número de linhas do código, resultado da implementação em uma linguagem interpretada.

Sob o estudo de caso na BHRX, o LASH obteve uma ótima performance estatística, i.e. coeficiente de Nash-Sutcliffe = 0,94. Além disso, a propagação de cheias via Onda Cinemática atenuou os picos e trasladou a onda de cheia de forma satisfatória e condizente com o tempo de concentração das sub-bacias da BHRX (> 1 dia). Como o hidrograma da BHRX é sazonal, ou seja, apresenta períodos bem marcados de enchente e vazante, a visualização da propagação da onda no rio é acentuada (Figura 2).

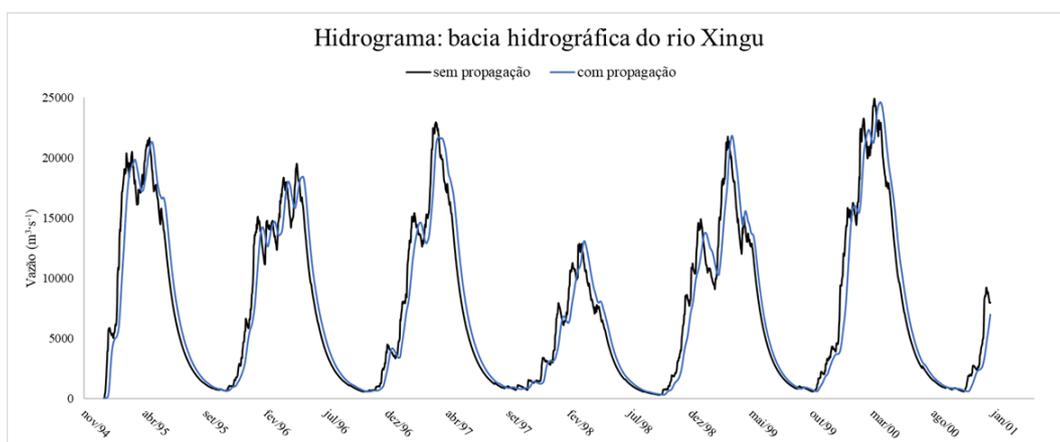


Figura 2 – Hidrograma da BHRX sob o aspecto da consideração da propagação da onda de cheia via método da Onda Cinemática.

A utilização do MATLAB® como ambiente de implementação do LASH possibilitou empregar funções prontas do algoritmo AMALGAM. No caso das versões em Delphi, por exemplo, o SCE-UA precisou ser integralmente escrito na linguagem. Essas premissas são importantes para elucidar a facilidade de utilização de uma linguagem como o MATLAB® em virtude de um maior aproveitamento de códigos aplicados à hidrologia.

No tocante à utilização do AMALGAM na BHRC, pode-se observar, mediante três funções objetivo avaliadas por meio de uma análise superficial, que o AMALGAM resultou em uma melhora das incertezas na modelagem em relação ao SCE-UA. Além disso, o LASH obteve uma boa performance (coeficiente de Nash-Sutcliffe = 0,76) quando do uso do AMALGAM, a qual foi ligeiramente superior em relação ao SCE-UA. A Figura 3 foi computada considerando as melhores populações retornadas pelo SCE-UA (28 populações) e pelo AMALGAM (50 populações).

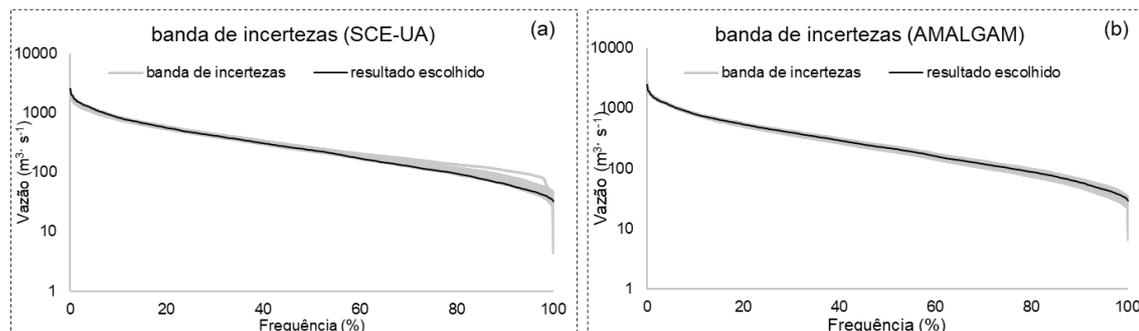


Figura 3 – Incertezas da modelagem associadas ao SCE-UA (a) e ao AMALGAM (b).

Um ganho importante na formulação do LASH em MATLAB® foi em relação ao tempo de processamento na calibração, com cerca de 2h para a BHRC (57 sub-bacias e tamanho de série histórica de 2.191 dias) e 3h para a BHRX (91 sub-bacias e tamanho de série histórica de 2.191 dias). Os resultados da formulação do LASH em MATLAB®, em termos de processamento, são animadores no que tange à utilização de um modelo hidrológico de longo termo e semi-distribuído, sendo um avanço considerável frente ao desenvolvimento do modelo.

4. CONCLUSÕES

Mediante o exposto, pode-se concluir que a implementação do modelo LASH em MATLAB® facilita a alteração, atualização e ampliação do *script*, sendo o grande gargalo da implementação do modelo em uma linguagem interpretada. Ademais, essa implementação permite a exploração de um produto final (o modelo hidrológico em *script*) passível de ser facilmente utilizado para analisar a hidrologia de bacias hidrográficas.

5. REFERÊNCIAS BIBLIOGRÁFICAS

- BESKOW, S. et al. Performance of a distributed semi-conceptual hydrological model under tropical watershed conditions. **Catena**, v. 86, p. 160–171, 2011.
- CALDEIRA, T. L. et al. LASH hydrological model: an analysis focused on spatial discretization. **Catena**, v. 173, p. 183-193, 2019.
- DUAN, Q.; SOROOSHIAN, S.; GUPTA, V. Effective and Efficient Global Optimization for Conceptual Rainfall-Runoff Models. **Water Resources Research**, v. 28, n. 4, p. 1015-1031, 1992.
- MELLO, C. R. et al. Development and application of a simple hydrologic model simulation for a Brazilian headwater basin. **Catena**, v. 75, n. 3, p. 235-247, 2008.
- PAIVA, R. 2007. Método de propagação baseado na Onda Cinemática.
- SANTOS, R. et al. Aprimoramentos computacionais da ferramenta ArcLASH. *In Anais XXVIII Congresso de Iniciação Científica da Universidade Federal de Pelotas*, 2019.
- SLATER, L. J. et al. Using R in hydrology: a review of recent developments and future directions. **Hydrology and Earth System Sciences**, v. 23, n. 7, p. 2939-2963, 2019.
- VRUGT; J. A.; ROBINSON, B. A. Improved evolutionary optimization from genetically adaptive multimethod search. **Proceedings of the National Academy of Sciences**, v. 104, n. 3, p. 708-711, 2007.