

Paralelização de métodos computacionais aplicados à análise das variações do teor de umidade de solos saturados e não saturados*

Edson Luis Padoin, Sergio Luis Dill, Roberto Schaedler Borges
DeTEC - Departamento de Tecnologia

Pedro Augusto Pereira Borges
DeFEM - Departamento de Física Estatística e Matemática

Universidade Regional do Noroeste do Estado do Rio Grande do Sul - UNIJUÍ
CP560, 98.700-000, Ijuí, RS, Brasil

{padoin, dill, beto, pborges}@unijui.tche.br

Resumo

O estudo do comportamento da água no solo é de grande interesse para as pesquisas em sistemas agroflorestais e agrícolas, já que o movimento dos nutrientes depende do movimento da água no solo. Nos projetos de irrigação, principalmente por gotejamento, a descrição de como a água se desloca é fundamental para saber a quantidade de água disponível para as plantas em cada profundidade do solo. A equação de Richards relaciona o teor de umidade do solo com o potencial total (soma dos potenciais matriciais, gravitacional, pressão e osmótico). Neste trabalho esta equação foi resolvida numericamente em duas dimensões, em coordenadas cilíndricas. Foi simulada a irrigação na fronteira superior, considerando um fluxo de entrada de água durante um tempo inicial e suspensão da irrigação nos instantes posteriores. Utilizou-se o método das diferenças finitas com diferenças centrais, com esquema explícito, devido à característica difusiva do fenômeno. Como a equação de Richards envolve o potencial total, o teor de umidade foi calculado com base na curva característica do solo, determinada experimentalmente. Foram realizados experimentos de irrigação de volumes cilíndricos de solo, nos quais foram feitas medidas do teor de umidade.

* Este trabalho foi realizado com o apoio UNIJUÍ, edital 06/2006 - VRPGPE.

Os resultados obtidos com a resolução seqüencial do problema descrevem a tendência do deslocamento da água do solo indicada pelos dados experimentais. Para obtenção de tais resultados demandaram-se grandes tempos de processamento mesmo utilizando-se matrizes de pequena ordem. Nesse sentido, tais implementações foram paralelizadas e executadas num aglomerado de computadores, onde se conseguiu aumentar a ordem do sistema e melhorar a precisão dos resultados.

1. Introdução

O estudo do comportamento da água no solo é de grande interesse para as pesquisas em sistemas agroflorestais e agrícolas, já que o movimento dos nutrientes depende do movimento da água no solo [1]. A descrição de como a água se desloca é fundamental nos projetos de irrigação, principalmente por gotejamento, para determinar a quantidade ótima de água necessária para o desenvolvimento das plantas em cada profundidade do solo.

O movimento da água no solo é modelado pela conhecida equação de Richards, cuja dedução é encontrada em vários trabalhos, tais como [1], [2], [3] e [4]. A não linearidade, devido à variabilidade da condutividade hidráulica em relação ao teor de umidade em cada ponto, inviabiliza uma solução analítica. Neste trabalho, a equação de Richards foi escrita em coordenadas cilíndricas e resolvida

numericamente usando o Método das Diferenças Finitas [5][6], para solos saturados e não saturados. O algoritmo possui dois fatores que explicam a grande demanda de poder de processamento e que tornam morosa a sua execução: 1º) A grande dependência de dados [7] nos comandos de decisão que verificam se cada ponto no interior do solo está saturado ou não saturado e 2º) O número ótimo de células da malha necessário para obter resultados com a precisão desejada. Visando a execução mais rápida da aplicação e no intuito de melhorar a precisão dos resultados na resolução dos sistemas de equações [8] [9], foi desenvolvida uma solução utilizando os recursos de processamento paralelo. Além disso, avalia-se também o número ótimo de nodos para cada malha.

Este trabalho é a primeira etapa da resolução do problema de movimento da água no sistema solo-atmosfera-plantas. Tal problema implica na resolução de um sistema de equações diferenciais parciais, que relaciona o crescimento do sistema radicular das plantas com o movimento da água no solo. Para realizar as simulações é necessário usar um intervalo de tempo reduzido para contemplar a variação do teor de umidade do solo (poucos minutos), em tempos reais suficientemente grandes (meses) para ocorrer o crescimento das raízes. Isso significa executar o programa de solução do movimento da água no solo milhares de vezes. Nesta primeira etapa foi implementada somente a solução do problema do movimento da água no solo, sem a presença de raízes de plantas. Apresentam-se os resultados da solução paralela para a análise das variações do teor de umidade de solos saturados e não-saturados. Para desenvolver esta abordagem, o restante do artigo segue organizado da seguinte forma: a Seção 2 introduz o problema e apresenta o modelo matemático; em seguida, na Seção 3, descrevem-se a solução numérica desenvolvida, os resultados obtidos na execução deste algoritmo sequencial. A solução paralela, o ambiente de execução e a análise dos resultados obtidos são apresentados na Seção 4; por fim, a Seção 5 relata algumas conclusões obtidas até o momento e perspectivas futuras de trabalho.

2. Descrição do Problema e do Modelo Matemático

O problema abordado neste trabalho constitui-se da avaliação de desempenho de um algoritmo utilizando os recursos de processamento paralelo e da necessidade de obtenção de resultados mais precisos na resolução do modelo matemático. O algoritmo resolve o

problema do movimento da água em um tubo cilíndrico de solo, considerando os estados de solo saturado e não saturado [10][11]. Foi considerado um tempo de irrigação no círculo central da superfície do cilindro, com um fluxo de água pré-determinado (Figura 1). O movimento da água no solo ocorre devido à existência de gradientes de potencial total (ψ). Este potencial é a soma dos potenciais matricial (ψ_m) e gravitacional (ψ_g), para solos não saturados gravitacionais e de pressão (ψ_p) e gravitacional para os solos saturados. O potencial osmótico também interfere no movimento da água, mas não será considerado neste trabalho, pois está associado à presença de películas (por exemplo, raízes). Pela ação dos potenciais, a água movimentar-se no solo nas direções r e z , fazendo variar o teor de umidade em cada ponto e instante de tempo.

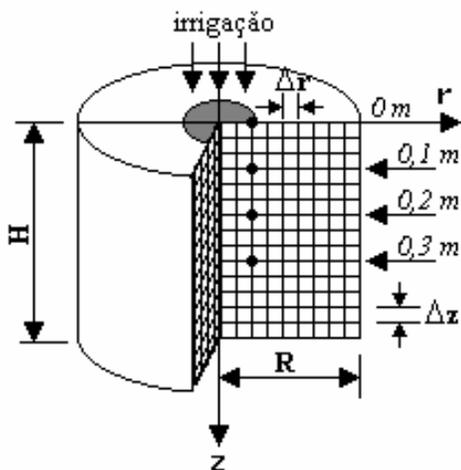


Figura 1 – Cilindro de solo submetido à irrigação com pontos de coleta de amostras e ilustração da malha utilizada na solução numérica.

A equação de Richards relaciona as variações do teor de umidade com as variações do potencial total em um solo isotrópico e homogêneo. Em coordenadas cilíndricas e para um problema de eixo simétrico esta equação tem a forma:

$$\frac{\partial \theta}{\partial t} = \frac{1}{r} \frac{\partial}{\partial r} \left[K_r(\theta) r \frac{\partial \Psi}{\partial r} \right] + \frac{\partial}{\partial z} \left[K_z(\theta) \frac{\partial \Psi}{\partial z} \right] \quad (1)$$

onde:

θ é o teor de umidade volumétrico do solo (adimensional)

Ψ é o potencial total (Pa)
 r e z são as variáveis espaciais (m)
 t é o tempo (s), e
 K_r e K_z são as condutividades hidráulicas nas direções r e z , respectivamente (m^2/sPa).

As condições de contorno que simulam as condições do experimento são:

$$\frac{\partial \Theta}{\partial r}(r, 0, t) = \Phi_i \text{ para } 0 < r \leq r_i \text{ e } t \leq t_i \quad (2)$$

$$\frac{\partial \Theta}{\partial r}(r, 0, t) = 0 \text{ para } r_i < r < R \text{ se } t \leq t_i \text{ e } 0 < r < R \text{ se } t > t_i \quad (3)$$

$$\frac{\partial \Theta}{\partial z}(r, H, t) = 0 \text{ para } 0 < r < R \text{ e } t > 0 \quad (4)$$

$$\frac{\partial \Theta}{\partial r}(R, z, t) = 0 \text{ para } 0 < z < H \text{ e } t > 0 \quad (5)$$

$$\Theta(r, z, 0) = \Theta_0 \text{ para } 0 < r < R \text{ e } 0 < z < H. \quad (6)$$

onde:

R é o raio do tubo (m)
 H é a altura do tubo (m)
 r_i é o raio de irrigação (m)
 t_i é o tempo de irrigação (s)
 Φ_i é o fluxo de irrigação (m^3/s).

Foram realizados experimentos de irrigação e monitoramento do teor de umidade no cilindro de solo na direção vertical (Figura 1). O cilindro de solo foi marcado na superfície, segundo quatro raios perpendiculares e nos planos definidos pelo deslocamento vertical de cada um deles, foram realizadas as coletas de solo em tempos diferentes e nos pontos indicados na Figura 1. As amostras de solo foram coletadas com um tubo de vidro e o teor de umidade foi medido utilizando o método gravitacional. Os resultados das medições do teor de umidade são apresentados pela Figura 3.

3. Solução Numérica

Foi usado o Método das Diferenças Finitas com diferenças centrais, por se tratar de um problema praticamente difusivo, devido à reduzida velocidade com que a água se desloca no solo. O domínio de integração foi definido em função de r e z , sendo que os volumes considerados em cada célula correspondem a anéis de raio r_i , espessura Δr e altura Δz (Figura 1).

A simulação da irrigação foi realizada considerando um fluxo de água (Φ_i) na superfície do cilindro. Para implementar a simulação da irrigação na superfície, foi considerado um fluxo de água constante durante 30 minutos. O esquema de interpolação temporal escolhido foi o esquema explícito. Nesse esquema o cálculo de $\Theta(r, z, t)$ para cada ponto da malha depende somente dos valores de $\Theta(r, z, t - \Delta t)$ os quais são todos conhecidos. Esse fato possibilita dividir o domínio de integração e realizar o cálculo de forma independente por cada nodo computacional, desde que sejam conhecidos os valores de teor de umidade, ao menos, dos pontos vizinhos [10][11].

O problema de convergência e estabilidade do esquema explícito foi resolvido calculado o incremento Δt , depois de definido o número de células nas direções r e z e conseqüentemente incrementos Δr e Δz , de tal forma que os coeficientes das variáveis Ψ e Θ fossem todos positivos (regra da positividade dos coeficientes).[12]

3.1. O Algoritmo Sequencial

O algoritmo sequencial é composto por 5 blocos. O primeiro bloco é o de entrada de dados, onde estão os parâmetros característicos do solo e as dimensões do cilindro. O segundo bloco é da definição da malha de diferenças finitas, onde são definidos os Δt , Δr e Δz em função das dimensões do cilindro e das exigências de convergência do problema. O terceiro bloco define as condições de contorno do problema de acordo com as equações (2) a (6). O quarto bloco calcula o teor de umidade usando a equação (1) discretizada para cada nó interno da malha e refaz as condições de fronteira que dependem do tempo. O quinto e último bloco tem os comandos de saída de dados. O fluxograma do algoritmo é apresentado na Figura 2.

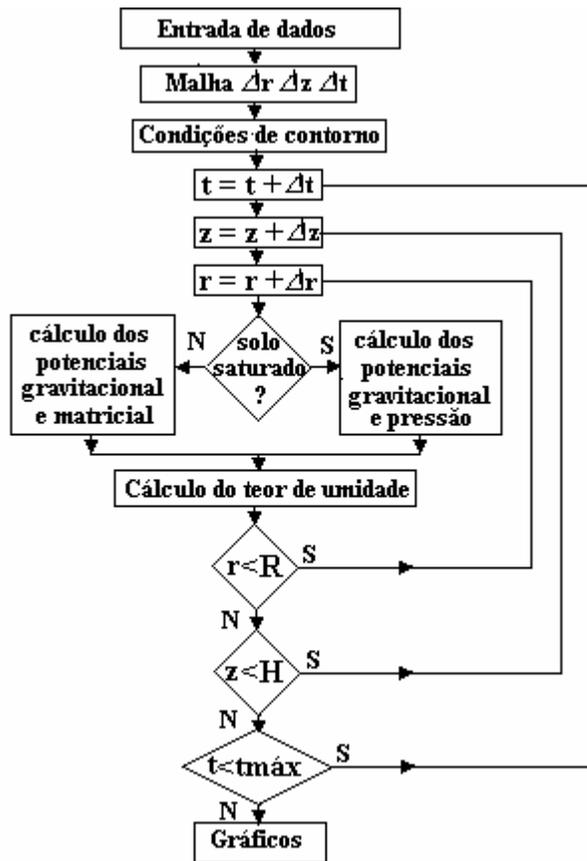


Figura 2 – Fluxograma do algoritmo seqüencial.

3.2. Resultados do Algoritmo Seqüencial

A Figura 3 mostra a comparação entre os dados calculados e medidos em função do tempo. Observa-se a forte influência do período de irrigação, tanto nas camadas da superfície como nas camadas interiores. O decréscimo repentino da curva referente à $z = 0m$ após o tempo de irrigação deve-se a interrupção do fluxo de água na superfície e a continuação da perda de água ocasionada pelo gradiente de umidade existente em relação às células vizinhas. As curvas calculadas reproduzem a tendência dos dados experimentais, pois, na medida em que o tempo aumenta, tendem para o mesmo valor do teor de umidade. Isto ilustra a influência do potencial matricial na homogeneização da distribuição da umidade no solo caracterizando um estado estacionário, quando o teor de umidade praticamente não varia significativamente em relação ao tempo.

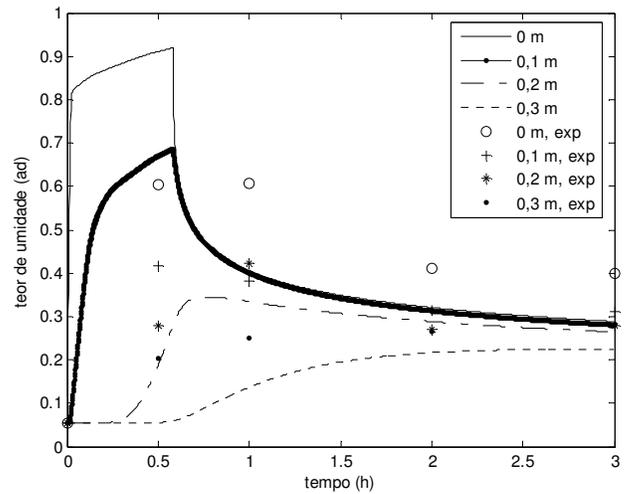


Figura 3 - Dados experimentais e resultados da aplicação do algoritmo seqüencial.

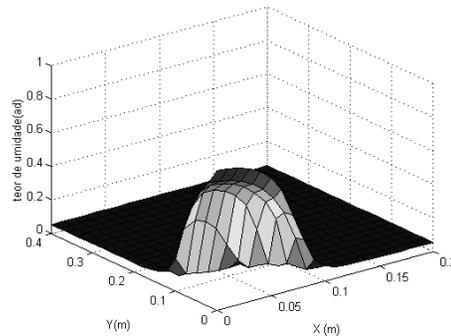


Figura 4 – Simulação da distribuição da umidade em uma seção do cilindro de solo: tempo = 1 000s.

A Figura 4 apresenta uma simulação (superfícies de umidade) para $t = 1000 s$, com os mesmos parâmetros e volume de irrigação da Figura 3. A superfície indica a evidente presença de irrigação no centro do cilindro e a distribuição de água mais concentrada próximo ao eixo do cilindro. Os resultados destes testes foram publicados em [10][11].

4. Solução Paralela

Os dados experimentais e resultados apresentados demandaram elevados tempos de processamento em uma malha de apenas 21 pontos. Torna-se difícil a exploração de resultados mais precisos com malhas maiores devido ao crescimento exponencial do tempo

de execução e da grande dependência de dados imposta pela aplicação. Assim, uma nova implementação foi desenvolvida em linguagem C, compilador *gcc* versão 3.3.5-5, utilizando a biblioteca PVM versão 3.4.5-3 para comunicação entre os nodos processadores. Com o objetivo de não só diminuir o tempo de processamento, mas também de aumentar a precisão dos resultados com o aumento do número de células na malha, diferentes alternativas de escalonamento foram pesquisadas.

Na implementação paralela do algoritmo utilizou-se uma programação do tipo mestre-escravo, pretendendo escalonar o trabalho entre os diversos nodos da máquina virtual formada pelo PVM. As escolhas por este modelo de programação e a utilização de PVM, se devem ao fato de que outros métodos matemáticos como Givens e Householder já foram paralelizados e publicados [13] utilizando tais requisitos o que permitirá a em breve a utilização destes modelos nesta pesquisa. A seguir são descritas as tarefas desempenhadas pelo nodo mestre e pelos nodos escravos.

- **Nodo-mestre**

A carga de processamento, representada por uma matriz $M \times M$, que representa os pontos do solo dentro do cilindro que é irrigado, é dividida igualmente entre cada nodo processador. O mestre faz a divisão da malha em linhas, divisão do domínio, e as envia juntamente com as informações de fronteira para os nodos escravos iniciarem o processamento em paralelo.

Devido à dependência de dados existente na aplicação torna-se necessário a comunicação entre os nodos processadores a cada iteração. Desta forma, o mestre recebe de cada escravo os respectivos pontos da malha calculados, com os novos valores, e atualiza a matriz enviando novamente aos escravos os subdomínios [14].

Uma iteração, passo temporal, é definida com o cálculo dos valores de cada célula da matriz que representa o sistema. Este cálculo iterativo é executado até que o número de iterações desejado seja atingido [15].

- **Nodos escravos**

Os processos escravos processam um conjunto de linhas da matriz. Cada nodo escravo recebe um subdomínio do mestre que coube a ele calcular, bem como os dados de fronteira. Assim cada nodo escravo calcula novos valores para os pontos da malha que possui com base no valor dos pontos vizinhos. Após efetuar o cálculo de todas as suas linhas, cada escravo

se comunica com o mestre enviando os seus pontos calculados. Os escravos ficam então aguardando o recebimento da nova malha para efetuarem novamente o cálculo.

4.1. Ambiente de Execução Paralela

Para a execução paralela da aplicação, utilizou-se um aglomerado homogêneo composto por 20 máquinas mono processadas Pentium Celeron R 2,6 GHz, com 256 MBytes de memória RAM, 128 KBytes de memória cachê L1, HD de 40 GBytes e adaptador de rede Fast Ethernet 100baseT. O sistema operacional de cada máquina é GNU/Linux (distribuição SuSE 9.3) com *kernel* versão 2.6.11.4. Aglomerado este formado pelas máquinas do laboratório de informática do Departamento de Tecnologia da UNIJUÍ [16].

4.2. Análise de Desempenho

Esta seção apresenta os resultados obtidos na execução do algoritmo paralelo. A análise do desempenho da implementação paralela foi realizada através da execução paralela com diferentes ordens de matrizes em diferentes números de nodos processadores, buscando obter a melhor relação de divisão de trabalho entre os nodos do aglomerado.

Na Tabela 1 apresenta-se os tempos de execução em segundos, o *speed-up* e a eficiência obtidos variando-se o número de nós do agregado e a ordem das matrizes. O algoritmo paralelo foi executado com matrizes de ordem 128, 256, 512 e 1024 sendo que o tempo representa a média de cinco execuções.

Analisando os tempos de execução do algoritmo paralelo apresentados na Tabela 1, observa-se que se obtiveram os menores tempos de execução utilizando-se 17, 13, 16 e 20 nodos para matrizes de ordem 128, 256, 512 e 1024 respectivamente. Sendo que, para matrizes de ordem 128 o tempo praticamente estabilizou, apresentando poucas variações a partir de 6 nodos. O mesmo é válido para as demais execuções, em 11, 13 e 18 respectivamente. Isso é justificado pela elevada dependência de dados do algoritmo que exige um grande número de comunicações entre os nodos do aglomerado. Tomando como base a execução sequencial do método para uma matriz de ordem 512 transcorreram 317 segundos, já com o método paralelo se alcançou o melhor desempenho, quando se reduzindo o tempo para 57 segundos, muito próximo da sexta parte do tempo sequencial, ou seja, um *speed-up* de 5,53 e uma eficiência de 35%.

Tabela 1 - Tempos de execução, speed-up e eficiência do algoritmo paralelo

nodos		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
128x128	Tempo	23,31	15,44	11,21	10,28	9,57	7,92	7,94	7,68	7,48	7,36	6,61	7,11	6,59	6,82	6,35	6,35	6,05	6,25	6,38	6,33
	128x128	1	1,510	2,080	2,267	2,437	2,945	2,936	3,036	3,116	3,169	3,527	3,279	3,540	3,419	3,674	3,671	3,854	3,728	3,655	3,682
	Eficiência	100%	76%	69%	57%	49%	49%	42%	38%	35%	32%	32%	27%	27%	24%	24%	23%	23%	21%	19%	18%
256x256	Tempo	80,69	47,13	34,50	26,85	26,81	23,96	23,77	22,87	23,02	19,46	18,88	21,47	17,27	19,41	18,91	17,50	19,36	18,05	18,05	18,14
	256x256	1	1,712	2,339	3,005	3,009	3,367	3,394	3,528	3,505	4,147	4,274	3,758	4,671	4,157	4,266	4,612	4,168	4,469	4,470	4,449
	Eficiência	100%	86%	78%	75%	60%	56%	48%	44%	39%	41%	39%	31%	36%	30%	28%	29%	25%	25%	24%	22%
512x512	Tempo	317,9	188,7	127,3	101,8	95,37	83,22	82,86	68,86	68,10	65,60	61,28	62,42	59,15	58,78	60,42	57,43	59,15	60,89	61,97	58,24
	512x512	1	1,685	2,498	3,124	3,334	3,820	3,837	4,617	4,669	4,846	5,188	5,093	5,375	5,409	5,262	5,536	5,375	5,222	5,130	5,459
	Eficiência	100%	84%	83%	78%	67%	64%	55%	58%	52%	48%	47%	42%	41%	39%	35%	35%	32%	29%	27%	27%
1024x1024	Tempo	1369	694,3	466,9	396,9	334,9	293,7	261,6	247,4	234,9	225,8	213,8	205,0	202,0	194,0	191,6	188,7	185,3	182,0	182,5	180,5
	1024x1024	1	1,971	2,931	3,449	4,087	4,660	5,231	5,532	5,826	6,060	6,401	6,675	6,774	7,056	7,143	7,254	7,386	7,521	7,500	7,580
	Eficiência	100%	99%	98%	86%	82%	78%	75%	69%	65%	61%	58%	56%	52%	50%	48%	45%	43%	42%	39%	38%

As Figuras 5, 6, 7 e 8 apresentam os tempos de execução do algoritmo paralelo para as matrizes de ordem 128, 256, 512 e 1024. Com base nestes resultados, pode-se analisar quanto o acréscimo de nodos processadores influencia no desempenho da execução paralela do algoritmo.

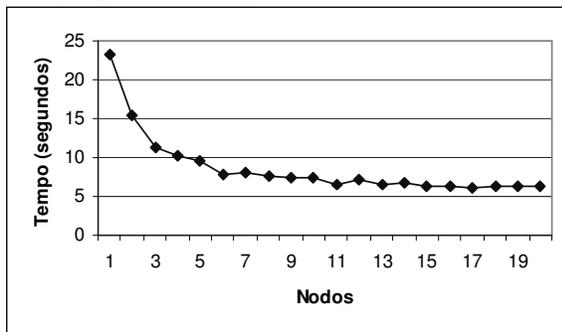


Figura 5 – Gráfico dos tempos de execução da implementação paralela com ordem 128

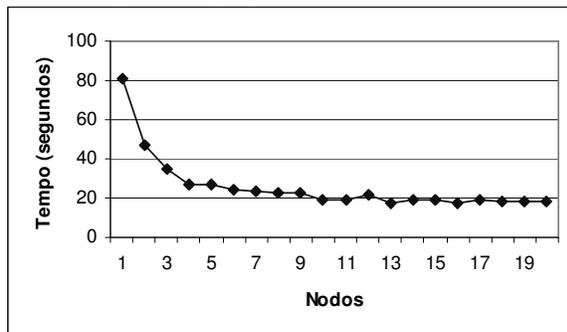


Figura 6 – Gráfico tempos de execução da implementação paralela com ordem 256

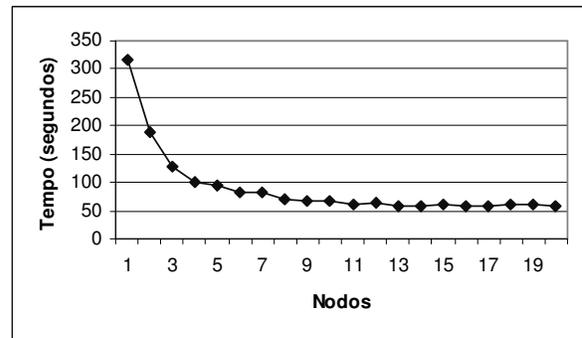


Figura 7 – Gráfico dos tempos de execução da implementação paralela com ordem 512

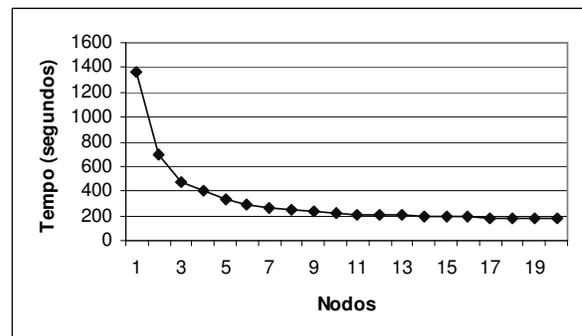


Figura 8 – Gráfico dos tempos de execução da implementação paralela com ordem 1024

Observando os gráficos pode-se concluir que as execuções do algoritmo paralelo para diferentes ordens de matriz possuem uma semelhança quanto ao desempenho. Ou seja, apresentam enormes ganho de desempenho com o acréscimo de poucos nodos processadores, sendo que, a partir deste ponto, observa-se uma pequena redução com tendência a estabilização. Isso, devido ao modelo de execução da aplicação, onde se tem um crescimento das comunicações relativo ao aumento do número de nodos processadores.

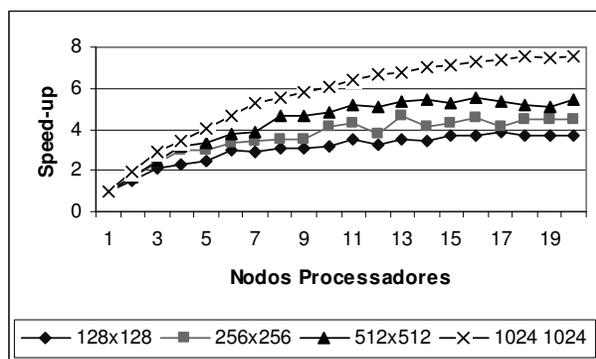


Figura 9 – Gráfico do *speed-up* da aplicação paralela executado com diferentes ordens de matriz

A Figura 9 apresenta a comparação do *speed-up* das execuções paralelas com matrizes de ordem 128, 256, 512 e 1024. Observa-se que para as diferentes execuções os valores de *speed-up* apresentam um comportamento muito semelhante, ou seja, em ambos apresentam uma elevação de acordo com o incremento da ordem do sistema.

5. Conclusão e Trabalhos Futuros

Foi desenvolvido um modelo matemático para calcular o teor de umidade de um solo submetido à irrigação. Foram realizados experimentos de medição do teor de umidade em diferentes pontos do solo em diferentes instantes de tempo. Os resultados do modelo reproduzem as tendências gerais do deslocamento da água no solo, indicadas pelos dados experimentais.

As simulações realizadas mostram o deslocamento da frente de umidade em uma seção longitudinal do cilindro para diferentes instantes de tempo e evidenciam a utilidade do modelo para estudar o comportamento da água no solo, considerando as influências das condições de fronteira.

O aumento da ordem do sistema, os erros de arredondamento, a falta de memória, entre outros fatores, muitas vezes limitam a resolução de sistemas ou até mesmo influenciam na qualidade das soluções encontradas. Assim, o desenvolvimento de programas paralelos é motivado não apenas pelo ganho de desempenho que se obtém com a execução em paralelo de rotinas, mas também pela possibilidade de exploração de novas alternativas para resolução de problemas. Neste trabalho apresentou-se uma análise de desempenho da versão paralela empregada na análise das variações do teor de umidade de solos saturados e não saturados usando PVM em um aglomerado de computadores.

Com a utilização de processamento paralelo conseguiu-se resolver o sistema em menores tempos de execução utilizando malhas compostas de um grande número de pontos o que conseqüentemente aumentou a precisão dos resultados. Tempo este que é limitado devido ao alto *overhead* de comunicação imposto pelo algoritmo. Os resultados obtidos permitem que trabalhos futuros possam ser desenvolvidos no intuito de validar novos testes na área de irrigação de solos sem a necessidade da execução de rotinas que demandem de grandes tempos de execução.

Como futuro trabalhos, pretende-se dar continuidade na pesquisa do problema de movimento da água no solo. Uma nova implementação será desenvolvida abordando não só o movimento da água, primeira parte da modelagem do sistema solo-atmosfera-planta, mas também o crescimento das raízes das plantas o que demandará um maior esforço computacional, além do desenvolvimento de novas estratégias de divisão de tarefas entre os nodos, justificando ainda mais o processamento paralelo. A atual implementação também está sendo aprimorada para execução no grid computacional.

Referências

- [1] K. Reichardt, A água em sistemas agrícolas. São Paulo: Editora Manoele Ltda, 1990.
- [2] P.L. Libardi, Dinâmica da água no solo. Piracicaba 1995, 497p.
- [3] P.L. Libardi, Dinâmica da água no sistema Solo/Planta/Atmosfera. Piracicaba: Gráfica CENA/USP, 1984, 531p
- [4] T. Miyazaki, S. Hasegawa and T. Kasubuchi, Water Flow in Soils. New York, Marcel Dekker, Inc., 1993.

- [5] G. D. Smith, 1985. Numerical Solution of Partial Differential Equations: Finite Difference Methods. 3ª ed. Clarendon Press Oxford, New York, USA.
- [6] B. Wilkinson and M. Allen. Parallel programming: techniques and applications using networked workstation and parallel computers. Prentice-Hall, Inc., 1999.
- [7] A. Y. Zomaya. Parallel and Distributed Computing Hand-book. McGraw-Hill, New York, 1996.
- [8] C.A. Hölgig, W. Krämer, and T.A. Divério. An accurate and efficient selfverifying solver for systems with banded coefficient matrix. In Parallel Computing: software Technology, Algorithms, architectures and Applications. Proceedings of the 10th ParCo Conference in Dresden, 2003, pages 283-290, Londres, 2004. Elsevier Science Publishers.
- [9] C.A. Hölgig, D. M. Cláudio and T.A. Divério. Obtendo alta exatidão em clusters: resolução de SELAS – um estudo de caso. In Workshop de Processamento Paralelo e Distribuído – WSPPD 2005, p. 33-38, Porto Alegre, RS, 2005.
- [10] Pedro A.P. Borges, Edson Luiz Padoin. Exemplos de Métodos Computacionais Aplicados a Problemas na Modelagem Matemática. In ERAD – Escola Regional de Alto Desempenho, 6. p.5-20. Ijuí – RS. 2006.
- [11] Borges, P. A. P., Coelho, G.C. e Buligon, S. L. Análise do comportamento da água em solos saturados e não saturados. In: XVIII Congresso Nacional de Matemática Aplicada e Computacional, CNMAC, Santo Amaro, SP, 2005.
- [12] C. R. Maliska, *Transferência de Calor e Mecânica dos Fluidos Computacionais*. Rio de Janeiro, livros Técnicos Científicos. 1995.
- [13] COPETTI, Alessandro, PADOIN, Edson L, Khatchatourian, O. Resolução Paralela de Sistemas de Equações Lineares Algébricos, In ERAD – Escola Regional de Alto Desempenho, 3. Santa Maria –RS. 2003.
- [14] Andrews, G., “Foundations of Multithread, Parallel, and Distributed Programming”. Addison Wesley, 2000.
- [15] M.V.Neves, T. Scheid, L.M. Schnorr, A.S. Charão. Integração de Garglia, libRastro e Pajé para monitoramento de aplicações paralelas. In Workshop de Computação de Alto Desempenho, 5. p.35-41, 2004.
- [16] R. Buyya. High Performance Cluster Computing: Architecture and Systems, volume 1. Prentice Hall, 1999.