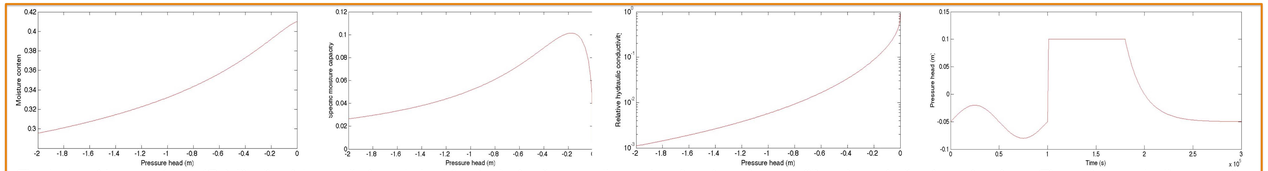
European Geosciences Union - *EGU General Assembly 2012* - **Poster # A328**

New challenges for Newton-type iteration schemes in Richards equation solvers: An example of infiltration driven by dynamic ponding conditions

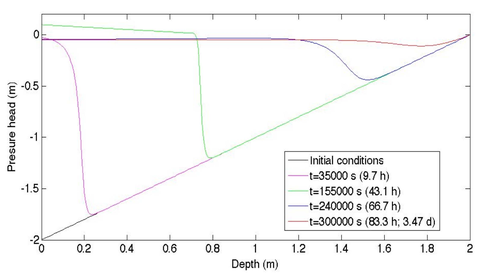
Claudio Paniconi *Institut National de la Recherche Scientifique – Centre Eau, Terre et Environnement (INRS-ETE), Université du Québec, Quebec City, Canada (claudio.paniconi@ete.inrs.ca)*

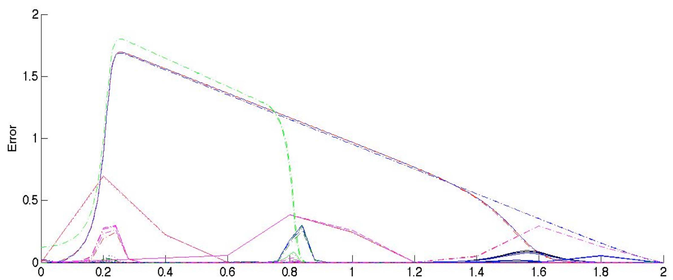
Mohammad Sayful Islam, Mario Putti *Dipartimento di Metodi e Modelli Matematici per le Scienze Applicate, Università di Padova, Padua, Italy*

As hydrological models become increasingly sophisticated (e.g., coupling with meteorological, ecological, or biogeochemical components) and are applied in ever more computationally demanding contexts (e.g., the many realizations that need to be generated in parameter estimation, uncertainty analysis, data assimilation, and scenario studies), the need for robust, accurate, and efficient codes is greater than ever. The Richards equation for subsurface flow is highly nonlinear and requires iterative schemes for its solution. These schemes have been the subject of much research over the past two decades *(e.g., Paniconi et al., 1991; Kirkland et al., 1992; Paniconi and Putti, 1994; Putti and Paniconi, 1994; Forsyth et al., 1995; Huang et al., 1996; Lehmann and Ackerer, 1998; Miller et al., 1998; Williams et al., 2000; Ross, 2003; Krabbenhøft, 2007; Crevoisier et al., 2009)*, but an effective all-purpose algorithm has thus far proven elusive. Ideally, rapid (quadratic as opposed to linear) and global (insensitive to initial guess) convergence is sought, as well as applicability over a range of conditions (dry soils, storm-interstorm simulations, geological heterogeneity, 3D domains with complex boundary conditions, etc). In addition to these challenges, the inclusion of surface/groundwater coupling *(e.g., Panday and Huyakorn, 2004; Kollet and Maxwell, 2006; Camporese et al., 2010)*, soil/vegetation interactions *(Hopmans and Bristow, 2002)*, data assimilation *(e.g., Paniconi et al., 2003; Camporese et al., 2009)*, and other processes and capabilities in advanced models can greatly impact the performance of an iteration scheme. In the case of integrated surface/subsurface flow modeling, for instance, convergence needs to be imposed on the coupling term in addition to the subsurface flow solution, atmospheric forcing becomes a dynamic process with highly variable rainfall and evaporation rates, and high ponding heads can form at the surface and add significantly to the computational burden. In this presentation we illustrate some of these new challenges by examining in detail the performance of the classic Picard and Newton iteration methods for a test case of dynamic forcing with high ponding heads imposed at the surface, and we give a preliminary assessment of a new scheme *(Casulli and Zanolli, 2010)* that uses a nested Newton algorithm based on a Jordan decomposition of the soil specific moisture capacity function.

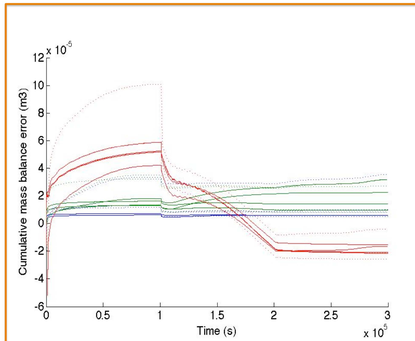


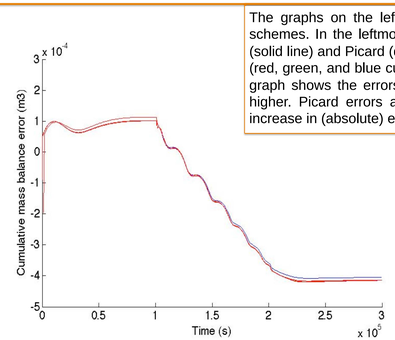
The test problem considers 1D infiltration into a soil characterized by the hydraulic properties and surface boundary condition shown in the 4 graphs above. The moisture retention curve is monotonic with a point of inflection that gives the moisture capacity function its typical shape. This feature is exploited in the scheme proposed by *Casulli and Zanolli (2010)*, whereby the Newton scheme is split between the increasing and decreasing regions of the moisture capacity curve. The Dirichlet boundary condition leads to significant ponding between 100000 and 200000 s (27.8 and 55.6 h), and as will be seen in the results, this type of boundary condition, prominent in coupled groundwater/surface water modeling, is a source of significant difficulty in the iterative schemes.





The initial conditions and pressure head solution profiles at 4 different times are shown in the left graph. The green profile, which falls within the ponding period, shows the excess water that forms at the soil surface and the rather sharp moisture front that is generated. In the right graph are plotted the errors at the 4 times shown in the left graph for a wide range of solutions. The rmse errors are computed against a reference solution that was generated with a 1000-layer discretization and strict time stepping and convergence criteria. The different solutions represent vertical discretizations of 10, 50, 250, and 500 layers (Δz = 20, 4, 0.8, and 0.4 cm, respectively), maximum time step sizes of Δt = 1000, 100, 10, and 1 s, and the Picard, Newton, and nested Newton schemes. As expected the errors are highest at the coarsest spatial and temporal discretizations, and the peak errors propagate with the moisture front that is moving downwards into the soil.





The graphs on the left show the mass balance behavior for the Picard and Newton schemes. In the leftmost graph the cumulative mass errors are plotted for the Newton (solid line) and Picard (dotted line) schemes for the 50, 250, and 500-layer discretizations (red, green, and blue curves, respectively) and for all time step sizes, while the rightmost graph shows the errors for the 10-layer case, which are about an order of magnitude higher. Picard errors are generally a little higher than Newton errors, and the sharp increase in (absolute) error during the ponding period can be noted.

In the graphs on the right the adaptive time stepping behavior of the Newton (top graph) and Picard (bottom graph) is shown for all vertical discretizations. Most striking here is the very different behavior between the Newton and Picard schemes during the ponding period. Whereas the Newton scheme is forced to take very small step sizes only at the very beginning and end of the ponding period, the Picard scheme needs to negotiate a wide range of step sizes throughout the ponding period, and indeed for the Δt_{max} = 1000 s case it never achieves this maximum value during ponding, for any of the vertical discretizations. Constraining an iteration scheme to take extremely small time steps for prolonged periods during a simulation can represent an enormous computational burden for subsurface solvers.

References

M. Camporese, C. Paniconi, M. Putti, & P. Salandin, "Ensemble Kalman filter data assimilation for a process-based catchment model of surface and subsurface flow," *Water Resour. Res.,* 45, W10421, 2009

M. Camporese, C. Paniconi, M. Putti, & S. Orlandini, "Surface–subsurface flow modeling with path-based runoff routing, boundary condition-based coupling, and assimilation of multisource observation data," *Water Resour. Res.,* 46, W02512, 2010

V. Casulli & P. Zanolli, "A nested Newton-type algorithm for finite volume methods solving Richards' equation in mixed form," *SIAM J. Sci. Comput.,* 32(4), 2255-2273, 2010

D. Crevoisier, A. Chanzy, & M. Voltz, "Evaluation of the Ross fast solution of Richards' equation in unfavourable conditions for standard finite element methods," *Adv. Water Resour.,* 32(6), 936-947, 2009

P. A. Forsyth, Y. S. Wu, & K. Pruess, "Robust numerical methods for saturated-unsaturated flow with dry initial conditions in heterogeneous media," *Adv. Water Resour.,* 18, 25-38, 1995

J. W. Hopmans & K. L. Bristow, "Current capabilities and future needs of root water and nutrient uptake modeling," *Adv. Agron.,* 77, 103-183, 2002

K. Huang, B. P. Mohanty, & M. Th. van Genuchten, "A new convergence criterion for the modified Picard iteration method to solve the variably saturated flow equation," *J. Hydrol.,* 178, 69-91, 1996

M. R. Kirkland, R. G. Hills, & P. J. Wierenga, "Algorithms for solving Richards' equation for variably saturated soils," *Water Resour. Res.,* 28(8), 2049-2058, 1992

S. J. Kollet & R. M. Maxwell, "Integrated surface–groundwater flow modeling: A free-surface overland flow boundary condition in a parallel groundwater flow model," *Adv. Water Resour.,* 29(7), 945-958, 2006

K. Krabbenhøft, "An alternative to primary variable switching in saturated–unsaturated flow computations, *Adv. Water Resour.,* 30, 483-492, 2007

F. Lehmann & P. Ackerer, "Comparison of iterative methods for improved solutions of the fluid flow equation in partially saturated porous media," *Transport in Porous Media,* 31(3), 275-292, 1998

C. T. Miller, G. A. Williams, C. T. Kelley, & M. D. Tocci, "Robust solution of Richards' equation for nonuniform porous media," *Water Resour. Res.,* 34(10), 2599-2610, 1998

S. Panday & P. S. Huyakorn, "A fully coupled physically-based spatially-distributed model for evaluating surface/subsurface flow," *Adv. Water Resour.,* 27, 361-382, 2004

C. Paniconi, A. A. Aldama, & E. F. Wood, "Numerical evaluation of iterative and noniterative methods for the solution of the nonlinear Richards equation," *Water Resour. Res.,* 27(6), 1147-1163, 1991

C. Paniconi & M. Putti, "A comparison of Picard and Newton iteration in the numerical solution of multidimensional variably saturated flow problems," *Water Resour. Res.,* 30(12), 3357-3374, 1994

C. Paniconi, M. Marrocu, M. Putti, & M. Verbunt, "Newtonian nudging for a Richards equation-based distributed hydrological model," *Adv. Water Resour.,* 26(2), 161-178, 2003

M. Putti & C. Paniconi, "Quasi-Newton methods for Richards' equation," In: *Computational Methods in Water Resources X, Volume 1,* pp. 99-106, Kluwer Academic, Dordrecht, The Netherlands, 1994

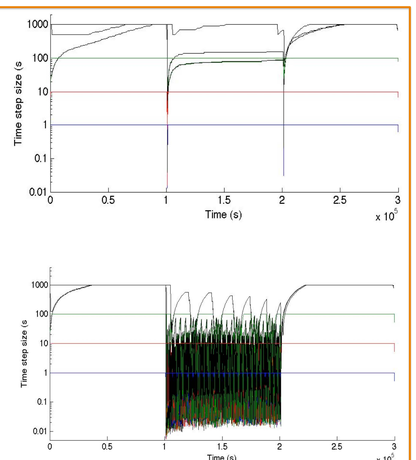
P. J. Ross, "Modeling soil water and solute transport: Fast, simplified numerical solutions," *Agron. J.,* 95, 1352-1361, 2003

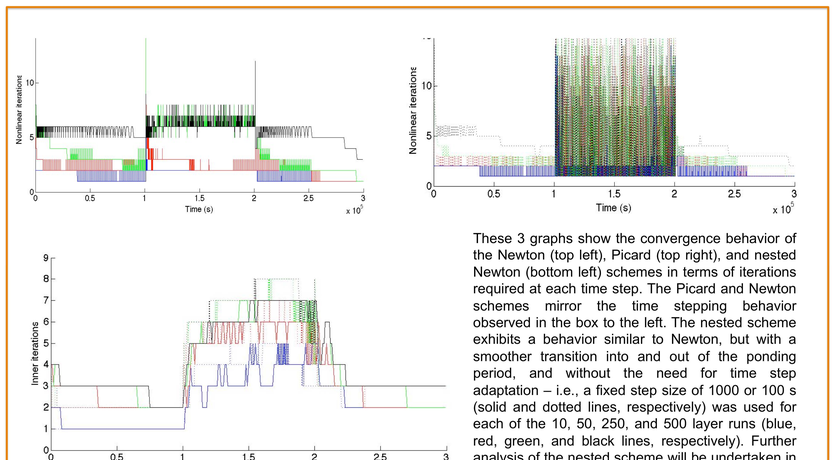
G. A. Williams, C. T. Miller, & C. T. Kelley, "Transformation approaches for simulating flow in variably saturated porous media," *Water Resour. Res.,* 36(4), 923-934, 2000

Δt_{max} (s)	1000				100				10				1			
Δz (cm)	MBE (m^3)	MBE (%)	# Time Steps	Average Δt (s)	MBE (m^3)	MBE (%)	# Time Steps	Average Δt (s)	MBE (m^3)	MBE (%)	# Time Steps	Average Δt (s)	MBE (m^3)	MBE (%)	# Time Steps	Average Δt (s)
	Avg NL Iter/Step	Avg Linear Iter/NL Iter	# Back Steps	# Solver Failures	Avg NL Iter/Step	Avg Linear Iter/NL Iter	# Back Steps	# Solver Failures	Avg NL Iter/Step	Avg Linear Iter/NL Iter	# Back Steps	# Solver Failures	Avg NL Iter/Step	Avg Linear Iter/NL Iter	# Back Steps	# Solver Failures
20	-4.14e-4	-8.35e0	2595	116	-4.14e-4	-8.35e0	4790	62.6	-4.15e-5	-8.38e0	30333	9.89	-4.15e-4	-8.38e0	300000	1.00
	3.24	7.99	153	0	2.43	6.49	149	0	1.70	3.67	45	0	1.01	2.00	0	0
4	-4.19e-6	-7.75e-2	19532	15.4	-2.58e-5	-4.79e-1	21361	14.0	-2.11e-5	-3.91e-1	42583	7.05	-2.13e-5	-3.96e-1	303006	0.99
	3.01	6.92	1224	0	2.93	6.65	1242	0	2.22	4.73	1084	0	1.43	3.00	367	0
0.8	3.18e-5	5.84e-1	119616	2.51	2.70e-5	4.97e-1	121014	2.48	8.12e-6	1.50e-1	136622	2.20	9.85e-6	1.82e-1	364601	0.82
	2.56	4.70	8113	0	2.55	4.68	8062	0	2.44	4.38	7911	0	1.65	3.09	6290	0
0.4	3.55e-5	6.52e-1	165027	1.82	9.85e-6	1.82e-1	166843	1.80	8.44e-6	1.56e-1	186242	1.61	4.95e-6	9.15e-2	403462	0.74
	2.44	4.29	11298	0	2.44	4.27	11314	0	2.37	4.02	11419	0	1.70	3.02	9564	0

These two tables present summary statistics for the Picard (above) and Newton (below) schemes run over a range of mesh discretizations (10, 50, 250, and 500 layers, i.e., Δz = 20, 4, 0.8, and 0.4 cm, respectively) and maximum time step sizes (Δt = 1000, 100, 10, and 1 s). The 8 performance indicators are: total volumetric and percentage mass balance error, total number of time steps and average step size, the average number of Newton or Picard iterations taken at each time step, the average number of iterations needed to solve the linear algebraic system at each Picard or Newton iteration, the number of back-stepping occurrences (failure of Picard or Newton to converge within a preset maximum number of iterations – 15 in these runs – so that the time step is repeated at a smaller step size), and the number of linear solver failures (only relevant for the Newton scheme, which generates a nonsymmetric algebraic system, and noteworthy for two non-recoverable failures shown below).

Δt_{max} (s)	1000				100				10				1			
Δz (cm)	MBE (m^3)	MBE (%)	# Time Steps	Average Δt (s)	MBE (m^3)	MBE (%)	# Time Steps	Average Δt (s)	MBE (m^3)	MBE (%)	# Time Steps	Average Δt (s)	MBE (m^3)	MBE (%)	# Time Steps	Average Δt (s)
	Avg NL Iter/Step	Avg Linear Iter/NL Iter	# Back Steps	# Solver Failures	Avg NL Iter/Step	Avg Linear Iter/NL Iter	# Back Steps	# Solver Failures	Avg NL Iter/Step	Avg Linear Iter/NL Iter	# Back Steps	# Solver Failures	Avg NL Iter/Step	Avg Linear Iter/NL Iter	# Back Steps	# Solver Failures
20	-4.05e-4	-8.17e0	357	840	-4.14e-4	-8.36e0	3016	99.5	-4.15e-4	-8.38e0	30000	10.0	-4.15e-4	-8.37e0	300000	1.00
	4.60	7.54	1	0	2.12	5.33	0	0	1.68	3.50	0	0	1.01	2.16	0	0
4	-1.65e-5	-3.05e-1	1281	234	-1.52e-5	-2.82e-1	3187	94.1	-2.09e-5	-3.87e-1	30149	9.95	-2.14e-5	-3.97e-1	300096	0.9997
	5.58	8.23	15	1	3.38	6.42	7	0	1.86	4.17	6	0	1.42	3.24	2	0
0.8	3.19e-5	5.87e-1	2237	134	2.24e-5	4.13e-1	3737	80.3	1.40e-5	2.59e-1	30247	9.92	9.74e-6	1.80e-1	300083	0.9997
	5.93	8.04	22	3	4.26	6.88	12	1	2.05	4.83	16	2	1.47	3.52	6	0
0.4	Divergent solution				Divergent solution				5.90e-6	1.09e-1	30266	9.91	5.84e-6	1.08e-1	300196	0.9993
									2.08	4.95	14	1	1.47	3.70	13	3





These 3 graphs show the convergence behavior of the Newton (top left), Picard (top right), and nested Newton (bottom left) schemes in terms of iterations required at each time step. The Picard and Newton schemes mirror the time stepping behavior observed in the box to the left. The nested scheme exhibits a behavior similar to Newton, but with a smoother transition into and out of the ponding period, and without the need for time step adaptation – i.e., a fixed step size of 1000 or 100 s (solid and dotted lines, respectively) was used for each of the 10, 50, 250, and 500 layer runs (blue, red, green, and black lines, respectively). Further analysis of the nested scheme will be undertaken in the coming months.