An investigation of a dual-porosity model for the simulation of unsaturated flow in a porous medium

Michael H.Brookes
September, 1994

Submitted to The University of Reading ,
Department of Mathematics,
in partial fulfillment of the requirements for the
Degree of Master of Science.

Acknowledgments

I would like to thank my supervisor Dr.W.L.Wood of The University of Reading for her advice and help during the writing of this dissertation.I would also like to thank Dr.A.Calver of The Institute of Hydrology for proposing this dissertation and for hydrological advice.

I acknowledge the financial support which I have received from EPSRC (formerly SERC).

Abstract

A 1-D dual porosity model using Richards' [Richards 1931] coupled non-linear parabolic equations are solved numerically with finite differences.This model is especially appropriate when modelling unsaturated ground water flow in fractured rocks or cracked soils.

Iteration was required to achieve the numerical solution due to the non-linear form of the parabolic equations.

Experiments were made to find the effects of varying parameters on the accuracy,on the convergence of iteration and on the computational effort required to achieve the numerical solution.

Some numerical examples are also included to show the hydraulic plausibility of the numerical results obtained.

Contents

Chapter 1

Dual porosity ground water flow

1.1 Introduction

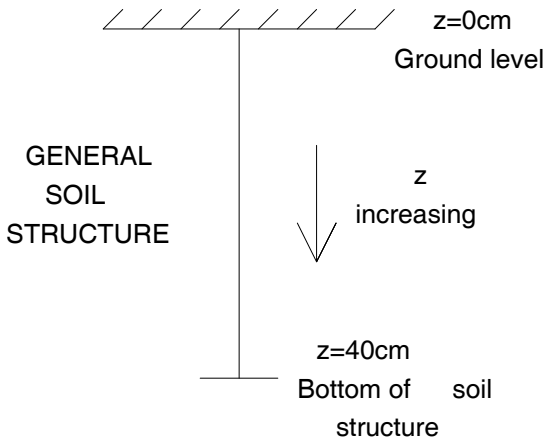

Figure 1.1: The domain of the model.

In this dissertation we will examine the model proposed in the 1993 paper of Gerke and van Genuchten.This model is a one-dimensional dual-porosity model which can be used to study variably saturated water flow in structured soil or fractured rock.The domain of the soil is shown in figure 1.1 .

The model consists of superposing two continua at macroscopic level.The two continua are a macropore or fracture pore system and a less permeable matrix pore system.Mobile water exists in both pores.There is an assumption of no horizontal flow taking place since the rock(or soil structure)is assumed to be horizontally homogeneous.

Richard's equation[Richards 1931]describes the variably saturated water flow in both pore systems.The transfer of water between the two pore regions is simulated by means of first-order equations governing the rate.

The model results in two coupled systems of nonlinear partial differential equations which can be solved numerically using Galerkin Finite Element Methods and mass lumping in space(as in [Milly 1985] as 'L1').The time stepping was achieved by the fully implicit θ method.

The new approach which we are going to introduce here uses the same basic model as used by Gerke and van Genuchten in their 1993 paper,but solves the coupled system of nonlinear partial differential equations by finite difference schemes.In space the integration method [Wood 1993] is used and in time we use the fully implicit θ method.This approach is different to that taken by Gerke and van Genuchten because a different form of mass lumping is implicitly used(as in[Milly 1985]'L2').

Further,when the numerical solution is available,we will use the program to experiment to try to find the best ways of producing the results.This means trying to find the fastest,most accurate and most reliable way a numerical solution may be obtained given general initial, and boundary conditions of the problem.

Further still,we will also try to investigate the dependence of the numerical solution on different types of initial, and boundary conditions and the ease with which different specifications of numerical solutions may be obtained.

Finally we will conclude this dissertation with some simple numerical examples.

Since we use the same model as Gerke and van Genuchten used (in their 1993 paper) ,we must explain that model in more detail.This we do in the next section,section 1.2 .

1.2 The Model of the Problem

The equations of the dual-porosity model are

$$c_f \frac{\partial h_f}{\partial t} = \frac{\partial}{\partial z}(K_f \frac{\partial h_f}{\partial z} - K_f) - \frac{\gamma_w}{w_f} \tag{1.1}$$

$$c_m \frac{\partial h_m}{\partial t} = \frac{\partial}{\partial z}(K_m \frac{\partial h_m}{\partial z} - K_m) + \frac{\gamma_w}{1 - w_f} \tag{1.2}$$

where c_m and c_f are the specific soil water capacities of the matrix and fracture pores respectively. h_f and h_m are the pressure heads of the fracture and matrix pores respectively.Also K_f and K_m are the hydraulic conductivities of fracture and matrix pores. γ_w is a term representing water transfer from fracture to matrix pores. The term w_f represents the volume of fracture pores as a proportion of total volume. Finally t is time and z is spatial distance(measured downwards with z=0 being ground level). 1.1 and 1.2 are forms of Richard's equation[Richards 1931].

The soil retention functions θ_m and θ_f are defined to be the amount of water present in a representative elementary volume of the matrix or fracture pores(respectively), divided by that representative elementary volume.

They are defined here as in [Gerke and van Genuchten 1993]in terms of $\theta_{r(m/f)}$, $\theta_{s(m/f)}$, $\alpha_{(m/f)}$, $h_{(m/f)}$, $n_{(m/f)}$ and $m_{(m/f)}$ as follows

$$\theta_m = \theta_{rm} + (\theta_{sm} - \theta_{rm})[1 + |\alpha_m h_m|^{n_m}]^{-m_m} \tag{1.3}$$

$$\theta_f = \theta_{rf} + (\theta_{sf} - \theta_{rf})[1 + |\alpha_f h_f|^{n_f}]^{-m_f} \tag{1.4}$$

where θ_{rf} , θ_{rm} are residual soil water retention constants for fracture or matrix pores. Similarly θ_{sf} , θ_{sm} are the saturated soil retention constants for fracture or matrix pores.

m_m , α_m , n_m or m_f , α_f , n_f are empirical constants for the matrix or fracture pores respectively.

The values of the specific soil water capacities c_m , c_f can be approximated to be the gradient of the soil retention functions θ_m or θ_f with respect to pressure heads h_m or h_f . That is;

$$c_m = \frac{d\theta_m}{dh_m} \tag{1.5}$$

$$c_f = \frac{d\theta_f}{dh_f} \tag{1.6}$$

The hydraulic conductivity of the matrix or fracture pores(K_m or K_f) can be thought of as representing the ease(or difficulty)with which water flows through their structures for a given pressure head gradient.This is shown by Darcy's Law [Darcy 1856] .

$$q_m = -K_m(\frac{\partial h_m}{\partial z} - 1) \tag{1.7}$$

$$q_f = -K_f(\frac{\partial h_f}{\partial z} - 1) \tag{1.8}$$

where q_m or q_f represent the downward flux of water flow in the matrix or fracture pores.

Gerke and van Genuchten(in their 1993 paper)use the van Genuchten formulae [van Genuchten 1980] for the hydraulic conductivity of the matrix and fracture pores (K_m or K_f)in terms of their hydraulic conductivity at saturation,as follows

$$K_m(S_{em}) = K_{sm} S_{em}^l [1 - (1 - S_{em}^{1/m_m})^{m_m}]^2 \tag{1.9}$$

$$K_f(S_{ef}) = K_{sf} S_{ef}^l [1 - (1 - S_{ef}^{1/m_f})^{m_f}]^2 \tag{1.10}$$

where K_{sm} or K_{sf} is the hydraulic conductivity at saturation of the matrix or fracture pores.The effective saturation parameters S_{em} and S_{ef} are defined in terms of the water content functions

$$S_{ef} = \frac{\theta_f - \theta_{rf}}{\theta_{sf} - \theta_{rf}} \tag{1.11}$$

$$S_{em} = \frac{\theta_m - \theta_{rm}}{\theta_{sm} - \theta_{rm}} \tag{1.12}$$

Boundary and initial conditions are necessary to solve equations 1.1 , 1.2 and these will be discussed in Chapter 2.

As noted by Gerke and van Genuchten[1993] the above model needs modification when the water downflow at the surface(caused by rainfall-for example)is larger than the matrix pores can absorb by themselves.If the water downflow is not so large that the entire absorption capacity of the soil is exceeded then a solution is

still possible.

In this case a positive pressure head builds up above the matrix pores.From this pressure head it is possible to find the pressure heads in the matrix and fracture pores at the surface of the soil using the equations 1.7 and 1.8 and the equation

$$q = w_f q_f + (1 - w_f) q_m \tag{1.13}$$

1.13 is derived as follows;

Fracture or matrix fluxes can be defined in the following equations:-

$$q_f = \frac{Q_f}{A_f} \tag{1.14}$$

$$q_m = \frac{Q_m}{A_m} \tag{1.15}$$

where q_f or q_m are the fluxes of water flowing through the fracture or matrix pores. Q_f or Q_m are the volumes of water(per unit time)flowing through the given areas A_f or A_m or the fracture or matrix pores.

The general flux of water, q, in the general soil structure can be described in terms of Q_f , Q_m , A_f and A_m as follows;

$$q = \frac{Q_f + Q_m}{A_f + A_m} \tag{1.16}$$

Now we are re-introducing the volumetric weighting factor w_f which is equal to the volume of fracture pores(v_f) as a proportion of total volume (v) in the following equation.

$$w_f = \frac{v_f}{v} \tag{1.17}$$

A_f and A_m may be defined in terms of A (the area of general soil structure perpendicular to the direction of flow of water(ie.horizontally))as follows

$$A_f = w_f A \tag{1.18}$$

$$A_m = (1 - w_f) A \tag{1.19}$$

This is valid if we assume (as implied by 1.17)

$$w_f (= \frac{v_f}{v}) = \frac{A_f}{A} \tag{1.20}$$

and we also assume

$$A_f + A_m = A \tag{1.21}$$

So from 1.16 we obtain

$$q = \frac{Q_f}{A_f + A_m} + \frac{Q_m}{A_f + A_m} \tag{1.22}$$

So using 1.21

$$q = \frac{Q_f}{A} + \frac{Q_m}{A} \tag{1.23}$$

Now using 1.18

$$q = \frac{w_f Q_f}{A_f} + \frac{Q_m}{A} \tag{1.24}$$

Now using 1.19

$$q = \frac{w_f Q_f}{A_f} + \frac{Q_m(1 - w_f)}{A_m} \tag{1.25}$$

ie.from 1.14 and 1.15 we get equation 1.13 .

Chapter 2

Numerical Solution of the Model

2.1 Spatial discretizion

The numerical solution of 1.1 and 1.2 will be by finite differences spatially.The derivation of this method is by the integration method[?]and is as follows;

From 1.1 and 1.2

$$0 = \frac{\partial}{\partial z}(K_f \frac{\partial h_f}{\partial z} - K_f) - \frac{\gamma_w}{w_f} - c_f \frac{\partial h_f}{\partial t} \tag{2.1}$$

$$0 = \frac{\partial}{\partial z}(K_m \frac{\partial h_m}{\partial z} - K_m) + \frac{\gamma_w}{w_m} - c_m \frac{\partial h_m}{\partial t} \tag{2.2}$$

(since $w_m + w_f = 1$, by definition)

Now we use

$$\gamma_w = \alpha_w^* K_a (h_f - h_m) \tag{2.3}$$

(as stated in [Gerke and van Genuchten 1993]) to eliminate γ_w from 2.1 and 2.2. Also multiplying 2.1 by w_f and 2.2 by w_m gives

$$0 = w_f \frac{\partial}{\partial z}(K_f \frac{\partial h_f}{\partial z} - K_f) - w_f c_f \frac{\partial h_f}{\partial t} - \alpha_w^* K_a h_f + \alpha_w^* K_a h_m \tag{2.4}$$

$$0 = w_m \frac{\partial}{\partial z}(K_m \frac{\partial h_m}{\partial z} - K_m) - w_m c_m \frac{\partial h_m}{\partial t} + \alpha_w^* K_a h_f - \alpha_w^* K_a h_m \tag{2.5}$$

So now dealing only with 2.4 we have

$$w_f \frac{\partial}{\partial z}(K_f \frac{\partial h_f}{\partial z} - K_f) = w_f c_f \frac{\partial h_f}{\partial t} + \alpha_w^* K_a h_f - \alpha_w^* K_a h_m \tag{2.6}$$

Integrating with respect to z from $z = z_{j-\frac{1}{2}}$ to $z_{j+\frac{1}{2}}$ (Fig 2.1)where $z_j = j\Delta z$ gives

$$\int_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} w_f \frac{\partial}{\partial z}(K_f \frac{\partial h_f}{\partial z} - K_f) dz = \int_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} (w_f c_f \frac{\partial h_f}{\partial t} + \alpha_w^* K_a h_f - \alpha_w^* K_a h_m) dz \tag{2.7}$$

Using an averaged value for the integrand on the right hand side of 2.7 , we have

$$\int_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} (w_f c_f \frac{\partial h_f}{\partial t} + \alpha_w^* K_a h_f - \alpha_w^* K_a h_m) dz = \Delta z([w_f c_f \frac{\partial h_f}{\partial t}]_{z_j} + [\alpha_w^* K_a h_f]_{z_j} - [\alpha_w^* K_a h_m]_{z_j}) \tag{2.8}$$

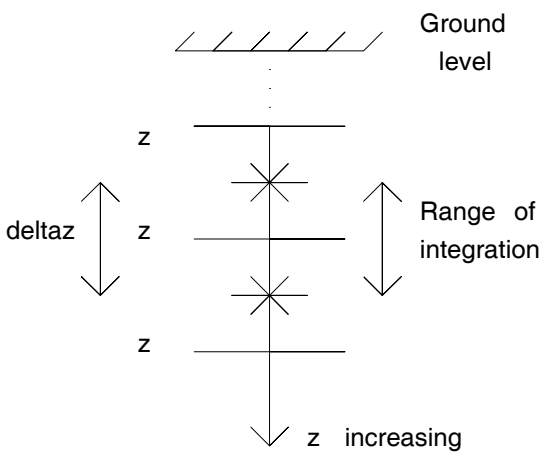

Figure 2.1: How 2.6 maybe spatially integrated.

Since w_f is a constant the left hand side of 2.7 may be integrated to give

$$\int_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} w_f \frac{\partial}{\partial z}(K_f \frac{\partial h_f}{\partial z} - K_f)dz = w_f([(K_f \frac{\partial h_f}{\partial z} - K_f)]_{z_{j+\frac{1}{2}}} - [(K_f \frac{\partial h_f}{\partial z} - K_f)]_{z_{j-\frac{1}{2}}}) \tag{2.9}$$

Now from 2.7 by using the approximation of 2.8 and the result of 2.9 we obtain

$$\begin{aligned} w_f([(K_f \frac{\partial h_f}{\partial z} - K_f)]_{z_{j+\frac{1}{2}}} - [(K_f \frac{\partial h_f}{\partial z} - K_f)]_{z_{j-\frac{1}{2}}}) \quad &= \quad \Delta z([w_f c_f \frac{\partial h_f}{\partial t}]_{z_j} \\ &+\alpha_w^*[K_a h_f]_{z_j} - \alpha_w^*[K_a h_m]_{z_j}) \end{aligned} \tag{2.10}$$

In the above equation it is obvious that some function values need to be evaluated mid-point between two spatial nodes.Since we only have values of these functions at the nodes,it is clear that some form of averaging will be required.In this numerical solution we will take the arithmetic average of the function values at the two nodes either side of the mid-point to be the value of the function at the mid-point.

From now on(for ease of notation)the subscript of a symbol will denote the number of its space-step and similarly the superscript will denote the number of its time-step.This will apply with the number of space-step increasing downwards(as z increases)and the number of time-step increasing with time.

From 2.10 and by approximating the space derivative(and by dividing by Δz)we find

$$\begin{aligned} w_f([K_f]_{j+\frac{1}{2}}(\frac{[h_f]_{j+1} - [h_f]_j}{\Delta z^2}) - \frac{[K_f]_{j+\frac{1}{2}}}{\Delta z}) \quad - \quad & w_f([K_f]_{j-\frac{1}{2}}(\frac{[h_f]_j - [h_f]_{j-1}}{\Delta z^2}) - \frac{[K_f]_{j-\frac{1}{2}}}{\Delta z}) \\ &= w_f[c_f \frac{\partial h_f}{\partial t}]_j + \alpha_w^*[K_a h_f]_j - \alpha_w^*[K_a h_m]_j \end{aligned} \tag{2.11}$$

Similarly we find that 2.5 gives

$$w_m([K_m]_{j+\frac{1}{2}}(\frac{[h_m]_{j+1}-[h_m]_j}{\Delta z^2})-\frac{[K_m]_{j+\frac{1}{2}}}{\Delta z}) \quad - \quad w_m([K_m]_{j-\frac{1}{2}}(\frac{[h_m]_j-[h_m]_{j-1}}{\Delta z^2})-\frac{[K_m]_{j-\frac{1}{2}}}{\Delta z})$$
$$= w_m[c_m\frac{\partial h_m}{\partial t}]_j+\alpha_w^*[K_a h_m]_j-\alpha_w^*[K_a h_f]_j \tag{2.12}$$

Now if we rearrange 2.11 then we obtain

$$[h_f]_{j+1}(\frac{w_f[K_f]_{j+\frac{1}{2}}}{\Delta z^2})-[h_f]_j(\frac{w_f[K_f]_{j+\frac{1}{2}}}{\Delta z^2} \quad + \quad \frac{w_f[K_f]_{j-\frac{1}{2}}}{\Delta z^2}+\alpha_w^*[K_a]_j)+[h_f]_{j-1}(\frac{w_f[K_f]_{j-\frac{1}{2}}}{\Delta z^2})$$
$$+[h_m]_j(\alpha_w^*[K_a]_j)=w_f[c_f\frac{\partial h_f}{\partial t}]_j \quad + \quad \frac{w_f[K_f]_{j+\frac{1}{2}}}{\Delta z}-\frac{w_f[K_f]_{j-\frac{1}{2}}}{\Delta z} \tag{2.13}$$

We will now represent 2.13 in a clearer form

$$A\underline{h}_f+B\underline{h}_m=C\frac{\partial \underline{h}_f}{\partial t}+\underline{F} \tag{2.14}$$

where $\underline{h}_f = [...,h_{f_{j-1}},h_{f_j},h_{f_{j+1}},...]^T$ and where $\underline{h}_m = [...,h_{m_{j-1}},h_{m_j},h_{m_{j+1}},...]^T$ also where

$$A=\begin{pmatrix} \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \frac{w_f[K_f]_{j-\frac{1}{2}}}{\Delta z^2} & -(\frac{w_f[K_f]_{j+\frac{1}{2}}}{\Delta z^2}+\frac{w_f[K_f]_{j-\frac{1}{2}}}{\Delta z^2}+\alpha_w^*[K_a]_j) & \frac{w_f[K_f]_{j+\frac{1}{2}}}{\Delta z^2} & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix}$$

$$B=\begin{pmatrix} \ddots & \ddots & \ddots \\ 0 & \alpha_w^*[K_a]_j & 0 \\ \ddots & \ddots & \ddots \end{pmatrix}$$

$$C=\begin{pmatrix} \ddots & \ddots & \ddots \\ 0 & w_f[c_f]_j & 0 \\ \ddots & \ddots & \ddots \end{pmatrix}$$

$$\underline{F}=\begin{pmatrix} \vdots \\ \frac{w_f[K_f]_{j+\frac{1}{2}}}{\Delta z}-\frac{w_f[K_f]_{j-\frac{1}{2}}}{\Delta z} \\ \vdots \end{pmatrix}$$

In all the above examples j increases or decreases as the row number of the matrix element increases or decreases.

When boundary conditions have been implemented we will find that the matrix A will become square.However boundary conditions are discussed more fully later on in this chapter.

2.2 Timelike discretization

The θ method is used to approximate the time derivative in equation 2.14.Therefore we obtain the following expression

$$\theta(A^{k+\frac{1}{2}}\underline{h}_f^{k+1}+B^{k+\frac{1}{2}}\underline{h}_m^{k+1})+(1-\theta)(A^{k+\frac{1}{2}}\underline{h}_f^{k}+B^{k+\frac{1}{2}}\underline{h}_m^{k}) = C^{k+\frac{1}{2}}(\frac{\underline{h}_f^{k+1}-\underline{h}_f^{k}}{\Delta t})+\underline{F}^{k+\frac{1}{2}} \tag{2.15}$$

In the above equation it is clear that timelike averaging between two time node points is necessary in order to evaluate the function value mid-way between two time nodes.This is done in exactly the same way as can be done spatially,ie.evaluating the function at both adjacent time nodes and then taking an arithmetic average.

It is well known(see [Wood 1993])that $\theta = 0.5$ tends to produce oscillations; $\theta = 1$ gives an unconditionally stable scheme with smoother results.Hence a value of $\theta = 1$ is used.

However using $\theta = 1$ has the disadvantage of requiring a matrix solver for each iteration of the numerical solution of 2.14

Discretising 2.14 as previously indicated gives the following equation:-

$$A^{k+\frac{1}{2}}\underline{h}_f^{k+1} + B^{k+\frac{1}{2}}\underline{h}_m^{k+1} = C^{k+\frac{1}{2}}(\frac{\underline{h}_f^{k+1}-\underline{h}_f^{k}}{\Delta t}) + \underline{F}^{k+\frac{1}{2}} \tag{2.16}$$

Using this form of time discretisation we find(from 2.13)that

$$\begin{aligned}
[h_f]_{j+1}^{k+1}\frac{w_f[K_f]_{j+\frac{1}{2}}^{k+\frac{1}{2}}}{\Delta z^2} &- [h_f]_j^{k+1}(\frac{w_f[K_f]_{j+\frac{1}{2}}^{k+\frac{1}{2}}}{\Delta z^2}+\frac{w_f[K_f]_{j-\frac{1}{2}}^{k+\frac{1}{2}}}{\Delta z^2}+\alpha_w^*[K_a]_j^{k+\frac{1}{2}}) \\
&+ [h_f]_{j-1}^{k+1}(\frac{w_f[K_f]_{j-\frac{1}{2}}^{k+\frac{1}{2}}}{\Delta z^2})+(\alpha_w^*[K_a]_j^{k+\frac{1}{2}})[h_m]_j^{k+1} \\
&= w_f[c_f]_j^{k+\frac{1}{2}}(\frac{[h_f]_j^{k+1}-[h_f]_j^{k}}{\Delta t})+\frac{[K_f]_{j+\frac{1}{2}}^{k+\frac{1}{2}}w_f}{\Delta z}-\frac{[K_f]_{j-\frac{1}{2}}^{k+\frac{1}{2}}w_f}{\Delta z}
\end{aligned} \tag{2.17}$$

We will now rearrange 2.17 to give

$$\begin{aligned}
[h_f]_{j+1}^{k+1}(\frac{w_f[K_f]_{j+\frac{1}{2}}^{k+\frac{1}{2}}}{\Delta z^2}) &- [h_f]_j^{k+1}(\frac{w_f[K_f]_{j+\frac{1}{2}}^{k+\frac{1}{2}}}{\Delta z^2}+\frac{w_f[K_f]_{j-\frac{1}{2}}^{k+1}}{\Delta z^2}+\alpha_w^*[K_a]_j^{k+\frac{1}{2}}+\frac{w_f[c_f]_j^{k+\frac{1}{2}}}{\Delta t}) \\
&+ [h_f]_{j-1}^{k+1}(\frac{w_f[K_f]_{j-\frac{1}{2}}^{k+\frac{1}{2}}}{\Delta z^2})+(\alpha_w^*[K_a]_j^{k+\frac{1}{2}})[h_m]_j^{k+1} \\
&= (\frac{-w_f[c_f]_j^{k+\frac{1}{2}}[h_f]_j^{k}}{\Delta t})+\frac{([K_f]_{j+\frac{1}{2}}^{k+\frac{1}{2}})w_f}{\Delta z}-\frac{([K_f]_{j-\frac{1}{2}}^{k+\frac{1}{2}})w_f}{\Delta z}
\end{aligned} \tag{2.18}$$

Similarly,by completing the same procedure(of rearranging,discretising in time and again rearranging)that we completed on 2.11 ,we find that from 2.12

$$
\begin{aligned}
[h_m]_{j+1}^{k+1}\left(\frac{w_m[K_m]_{j+\frac{1}{2}}^{k+\frac{1}{2}}}{\Delta z^2}\right) &- [h_m]_j^{k+1}\left(\frac{w_m[K_m]_{j+\frac{1}{2}}^{k+\frac{1}{2}}}{\Delta z^2}+\frac{w_m[K_m]_{j-\frac{1}{2}}^{k+1}}{\Delta z^2}+\alpha_w^*[K_a]_j^{k+\frac{1}{2}}+\frac{w_m[c_m]_j^{k+\frac{1}{2}}}{\Delta t}\right)\\
&+ [h_m]_{j-1}^{k+1}\left(\frac{w_m[K_m]_{j-\frac{1}{2}}^{k+\frac{1}{2}}}{\Delta z^2}\right)+(\alpha_w^*[K_a]_j^{k+\frac{1}{2}})[h_f]_j^{k+1}\\
&= \left(\frac{-w_m[c_m]_j^{k+\frac{1}{2}}[h_m]_j^k}{\Delta t}\right)+\frac{([K_m]_{j+\frac{1}{2}}^{k+\frac{1}{2}})w_m}{\Delta z}-\frac{([K_m]_{j-\frac{1}{2}}^{k+\frac{1}{2}})w_m}{\Delta z}
\end{aligned}
\tag{2.19}
$$

2.3 The General Solution

2.3.1 The form of the matrix problem

A five-diagonal matrix system can be formed by solving 2.18 and 2.19 simultaneously.Gerke and van Genuchten(in their 1993 paper)found the stability of the simultaneous solution to be superior to the stability exhibited when the equations were solved alternately.In the alternative approach the numerical solution is found by using 2.18 to update $\underline{h}_f^{k+1}$ and then using 2.19 to update $\underline{h}_m^{k+1}$ values.This process is then iterated until the updated values converge to a satisfactory tolerance.

The form of the simultaneous solution to the problem is

$$[R]^{k+\frac{1}{2}}[\underline{h}]^{k+1}=[\underline{G}]^{k+\frac{1}{2}} \tag{2.20}$$

where $[\underline{G}]^{k+\frac{1}{2}}$ is a column vector.The $c_{(f/m)}$ and $K_{(f/m)}$ terms have to be updated on each iterate since they have an implicit dependence on $[\underline{h}]^{k+1}$ .The exact form of $[\underline{G}]^{k+\frac{1}{2}}$ can be seen from the right hand side of 2.18 and 2.19 .

The term $[\underline{h}]^{k+1}$ now represents both the fracture and matrix values of the pressure heads and is shown below

$$[\underline{h}]^{k+1}=[...,[h_f]_{j-1}^{k+1},[h_m]_{j-1}^{k+1},[h_f]_j^{k+1},[h_m]_j^{k+1},[h_f]_{j+1}^{k+1},[h_m]_{j+1}^{k+1},...]^T \tag{2.21}$$

The structure of the matrix $[R]^{k+\frac{1}{2}}$ (as in 2.20)is non-trivial and is shown below

$$
\begin{pmatrix}
\ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots\\
\frac{w_f[K_f]_{j-\frac{1}{2}}^{k+\frac{1}{2}}}{\Delta z^2} & 0 & -[AA]_j^{k+\frac{1}{2}} & \alpha_w^*[K_a]_j^{k+\frac{1}{2}} & \frac{w_f[K_f]_{j+\frac{1}{2}}^{k+\frac{1}{2}}}{\Delta z^2} & 0 & 0 & 0\\
0 & \frac{w_m[K_m]_{j-\frac{1}{2}}^{k+\frac{1}{2}}}{\Delta z^2} & \alpha_w^*[K_a]_j^{k+\frac{1}{2}} & -[BB]_j^{k+\frac{1}{2}} & 0 & \frac{w_m[K_m]_{j+\frac{1}{2}}^{k+\frac{1}{2}}}{\Delta z^2} & 0 & 0\\
0 & 0 & \frac{w_f[K_f]_{j+\frac{1}{2}}^{k+\frac{1}{2}}}{\Delta z^2} & 0 & -[AA]_{j+1}^{k+\frac{1}{2}} & \alpha_w^*[K_a]_{j+1}^{k+\frac{1}{2}} & \frac{w_f[K_f]_{j+\frac{3}{2}}^{k+\frac{1}{2}}}{\Delta z^2} & 0\\
0 & 0 & 0 & \frac{w_m[K_m]_{j+\frac{1}{2}}^{k+\frac{1}{2}}}{\Delta z^2} & \alpha_w^*[K_a]_{j+1}^{k+\frac{1}{2}} & -[BB]_{j+1}^{k+\frac{1}{2}} & 0 & \frac{w_m[K_m]_{j+\frac{3}{2}}^{k+\frac{1}{2}}}{\Delta z^2}\\
\ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots
\end{pmatrix}
$$

where

$$[AA]_j^{k+\frac{1}{2}} = (\frac{w_f[K_f]_{j+\frac{1}{2}}^{k+\frac{1}{2}}}{\Delta z^2} + \frac{w_f[K_f]_{j-\frac{1}{2}}^{k+1}}{\Delta z^2} + \alpha_w^*[K_a]_j^{k+\frac{1}{2}} + \frac{w_f[c_f]_j^{K+\frac{1}{2}}}{\Delta t}) \quad (2.22)$$

and where

$$[BB]_j^{k+\frac{1}{2}} = (\frac{w_m[K_m]_{j+\frac{1}{2}}^{k+\frac{1}{2}}}{\Delta z^2} + \frac{w_m[K_m]_{j-\frac{1}{2}}^{k+1}}{\Delta z^2} + \alpha_w^*[K_a]_j^{k+\frac{1}{2}} + \frac{w_m[c_m]_j^{K+\frac{1}{2}}}{\Delta t}) \quad (2.23)$$

Notice there are more columns than rows in the matrix. Therefore boundary conditions(in terms of both fracture and matrix pressure heads)are required at both the top and bottom of the ground in order to solve for $[\underline{h}]^{k+1}$. The terms involving the boundary values are shifted to the right hand vector $[\underline{G}]$ (in 2.20)in order to make R (also in 2.20)square.

2.3.2 Method of iterating

Picard iteration is used due to the implicit dependence of $[\underline{G}]$ and $[\underline{R}]$ on $[\underline{h}]^{k+1}$ .We can accelerate the convergence of the iteration by selecting a suitable parameter ω in the following equation 2.24 .The updated value of $[\underline{h}]^{k+1,p+1}$ can be used as the best approximation(so far in the iteration)to the pressure head values $[\underline{h}]$ at the next time-step(k+1);ie.the program uses the following:

$$[\underline{h}]^{k+1,p+1} = [\underline{h}]^{k+1,p} + \omega([\underline{h}]^{k+1,p+1} - [\underline{h}]^{k+1,p}) \quad (2.24)$$

In equation 2.24 the first superscript refers to time-step of the pressure head values $[\underline{h}]$ and the second superscript(where used)denotes the iterate number.

The stopping condition for the iteration is when the maximum absolute difference between consecutive iterates is less than a pre-set tolerance.When this tolerance is met then time-stepping to the next time-step may proceed.

2.3.3 Method of time-stepping

Initially the value of $[\underline{h}]$ at the first time-step is approximated(to start the iteration)by assuming it to be identical to the initial values given.

Further timesteps are approximated in the first instance(to start the iteration)by using the formula

$$[\underline{h}]^{k+1,0} = [\underline{h}]^{k,p} + \Omega(\frac{\Delta t_k}{\Delta t_{k-1}})([\underline{h}]^{k,p} - [\underline{h}]^{k-1,q}) \quad (2.25)$$

where Δt_k is equal to the time stepped by the kth time-step. The above pressure head values are in the same notation as in 2.24 .Also in 2.25 ,we are simply using the superscripts p and q to represent the number of iterates required to achieve convergence at a given time.

The variable Ω is designed to be varied to increase the accuracy of the approximation of the pressure-head values at the first time-step(as ω was designed to accelerate the convergence of the iteration in equation 2.24).

2.3.4 Numerical values used in this dissertation

The various parameters which can be used were given in [Gerke and van Genuchten 1993]and are shown in Table 2.1 :-

	θ_r	θ_s	$\alpha.cm^{-1}$	n	l	$K_s.cm/day$	w	m
Fracture	0.0	0.5	0.1	2.0	0.5	2000.0	0.05	0.5
Matrix	0.10526	0.5	0.005	1.5	0.5	1.0526	0.95	$\frac{1}{3}$
Exchange Term	...	...	0.005	1.5	0.5	0.01	...	...

Table 2.1: Parameter values.

Spatial and time-steps are initially both uniform in this numerical solution,along with the depth of the soil region(which was always assumed to be 40cm).

The value of α_w^* ,as introduced in equation 2.3,is also constant.Its value is approximated in [Gerke and van Greuchten 1993] and is given in equation 2.26 .

$$\alpha_w^* = \frac{\beta}{a^2}\gamma \tag{2.26}$$

where β is a constant which depends on the geometry of the soil(equal to 3.0 for rectangular blocks), γ is an empirical constant for all soil types(its equal to 0.4), and a is the average size of the matrix blocks in the soil type.All these values are assumed to be constant for any one particular soil type.

The actual values of most of the functions which occur in the numerical solution can be evaluated using equations 1.9 , 1.10 , 1.11 , 1.12 , 1.3 , 1.4 and equations 1.5 , 1.6 of section 1.2.These functions are also graphically represented in figures 4.1 – 4.4 .

The only function which is introduced in chapter 2 and not explained in chapter 1 is the K_a function introduced in equation 2.3.This function is evaluated in the same way as for the K_m function,except that the K_s constant(for the exchange term)is much smaller(see table 2.1).The function K_a is also evaluated at a different pressure head value from K_m . K_a is evaluated at $[\overline{h}]_j$ where

$$[\overline{h}]_j = \frac{1}{2}([h_f]_j + [h_m]_j) \tag{2.27}$$

The numerical solution of the model may now be calculated.

Chapter 3

General Results

3.1 General properties of the numerical solution

3.1.1 Oscillation

No oscillation was experienced in the numerical results in time and in space.In addition it was noted that consecutive iterates(of a time-step)did not oscillate.The behaviour of the consecutive iterates not oscillating can probably be explained by the lumping of the mass matrix which has(implicitly)taken place in our solution(according to 'L2' in [Milly 1985])but which was done differently in the 1993 paper of Gerke and van Greuchten(according to 'L1' in [Milly 1985]).

In [Ouyang and Xiao 1994] there is a result stated regarding a linear parabolic problem which we have transformed to relevant variables,as follows;

$$\frac{\partial h}{\partial t} = \alpha \nabla^2 h(z,t) + f(z,t) \tag{3.1}$$

(where h represents the pressure head in either of the two pores and α is constant) Similar boundary and initial conditions to the ones we used in this problem are assumed.

The solution of equation 3.1 in space is assumed by finite element methods.This procedure is assumed to give the following equation;

$$[M]\dot{\underline{h}}(t) + [K]\underline{h}(t) = \underline{F} \tag{3.2}$$

(where now $\underline{h}$ is assumed to be the form of pressure heads as used in 2.21). Ouyang and Xiao assume the problem has been discretized in time by the θ method and then give the following condition (equation 3.3) as a sufficient condition of the non-oscillation of the numerical solution in *time*.The inequality of equation 3.3 is valid because $K_{ij} < 0$ if $i \neq j$ and $K_{ii} > 0$.

$$\max_{i,j}[\frac{M_{ij}}{-\theta K_{ij}}](i \neq j) \leq \Delta t \leq \min_{i}[\frac{M_{ii}}{(1-\theta)K_{ii}}] \tag{3.3}$$

Using our method of numerical discretization on equation 3.1 we may assume that the mass matrix M(as shown in 3.2)is in fact lumped and hence $M_{ij}(i \neq j)$

terms are zero.Further since we use the *fully* implicit θ method we may take θ to be equal to 1.Hence we conclude that condition 3.3 shows that our numerical solution on 3.1 will not oscillate in *time* when discretized by the numerical method we have used.

This result on our method of discretization of equation 3.1 applies to oscillation of the numerical solution in time but does not apply to the oscillation in *space* or to the oscillation of the iterates which may occur during the iteration of a time-step.However we conjecture that the result of [Ouyang and Xiao 1994] may suggest an explanation for the non-oscillation of our numerical solution(to equations 1.1 and 1.2)in *time* which we observe in this dissertation.Further investigation is required to confirm this conjecture.

3.1.2 Stability

Stability is not an issue in the numerical solution because a fully implicit difference scheme is used.

3.1.3 Convergence

By convergence we mean the ability of the pressure head values to converge to a new value for the new time-step,as the iterations continue.If the consecutive pressure head values begin to diverge,then convergence is unlikely.

To start with a fixed time and space-step is used,but later it was found to be beneficial to have a varying time-step.But in this section we shall assume the time-step to be fixed.

Often the only criterion for choosing the space and time-step was whether the iterative solution would converge.Decreasing the space-step(for a fixed overall soil depth)was found to have a detrimental effect on the convergence of the solution and decreasing the time-step was found to have a beneficial effect on the convergence(these results are also backed up by some of the results of section 3.6)

In the paper [Neuman 1973] it is suggested that the lumping of the mass matrix is necessary for convergence in an unsaturated flow.Indeed both Gerke and van Grenchten and the author did lump their mass matrices.

However Wood and Calver in their 1990 paper concluded that the distributed mass matrix should be used in saturated-unsaturated subsurface flow because it gives increased accuracy.This conclusion may be inappropriate in our case because we are dealing with unsaturated flow.

3.2 Applying rainfall

3.2.1 Boundary conditions

We briefly discussed boundary conditions in chapter 2.However it is perhaps appropriate for us to comment some more on how the boundary conditions were implemented in this numerical solution.

Usually Neumann boundary conditions were implemented.Therefore fictitious points had to be created in order to achieve these Neumann conditions.Strictly speaking these fictitious points are not within the soil structure and are not recorded in any results we show in future sections.

Upper boundary conditions

Using equations 1.7 and 1.8 these rainfall fluxes can be turned into pressure head gradients.The evaluation of the hydraulic conductivity term in 1.7 and 1.8 is at the point just below the fictitious point.

Once the pressure head gradient has been evaluated then the pressure head at the space point two space-steps below the fictitious point is changed by an amount in accordance with the gradient of the pressure head,until it reaches the value to be assigned at the fictitious point.This is shown in figure 3.1 .

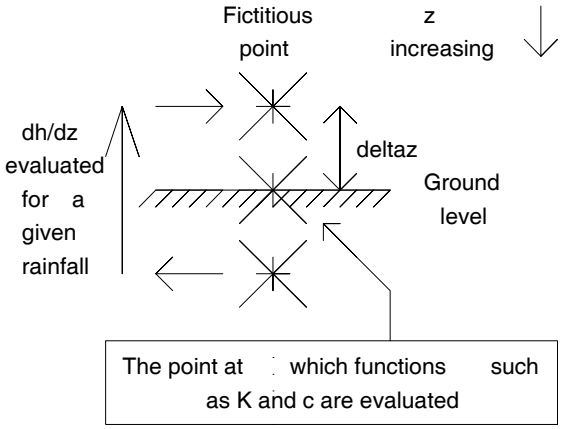

Figure 3.1: How upper boundary conditions are evaluated.

Later in this dissertation we may refer to'no flow'boundary conditions which merely means imposing a zero velocity of rainfall at the upper boundary.

Lower boundary conditions

These are almost always'free flow'boundary conditions.They are said to represent the unimpeded release of water as it exits from the soil structure(into a water table,for example).

These boundary conditions are simply implemented by making the value at the fictitious point (which is one vertical space-step below the lower boundary)equal that at the space node *on* the lower boundary.

More generally,the velocity of the flux of rainfall formed the upper boundary condition, and the lower boundary condition remained'free flow'(as before).

3.2.2 General properties

Generally the number of iterates required for each time-step(for fixed time-step)varied from problem to problem and varied as the fixed time-step varied for each prob-

lem.However it seemed that the number of iterates required(for a fixed time-step)seemed to diminish as the numerical solution tended towards the steady-state solution,or increase as the solution tended towards a saturated solution.Results backing-up some of these conclusions will be provided later.

3.3 Introducing the rainfall gradually

We now carried out a numerical example to see if(in this case)there was any benefit to be had by'running the problem in'as the flow of water at the top of the soil is introduced.

In general we would expect only a marginal difference unless the flow is introduced so slowly that the numerical solution has had time to move significantly closer to the steady-state solution,before the full impact of the flow is felt.Again the number of iterates required(per time-step)could form the basis of an indication as to when it is safe to add extra rainfall.

However,in the following example we do demonstrate the beneficial effects of introducing the flow gradually.

Numerical example

Firstly we introduced a rainfall of 3cm/day into the matrix pores('no flow'boundary conditions were imposed on the fracture).Both the fracture and matrix pores have an initial uniform pressure head of -1000cm(This very dry initial condition is used by Gerke and van Genuchten in their 1993 paper).A'Free flow'boundary condition was imposed at the lower end of the soil structure in both media.We set $\Delta z = 0.2$cm and set $\Delta t = 10^{-5}$days.We also set the transfer term to represent rectangular blocks sized 1cm(to let some of the water flow into the fracture)and the tolerance of the iteration to be 10^{-3}cm .The program initially converged in about 10 iterates(per time-step)and then it lost convergence after the fourth time-step.

Secondly we repeated the above experiment-but now introducing the rainfall gradually(by slowly increasing the flow(uniformly)over 100 timesteps until 3cm/day had been reached).The program did not loose convergence and iterated in about 3 iterates(per time-step).The program continued to run after the 3cm/day rainfall had begun to be added(although the number of iterates(per time-step)began to increase as the soil became more saturated).

This demonstrates the beneficial effects of a gradual introduction of rainfall.However we have picked a rather unusual example because,in the first case,the numerical solution was very close to converging and so it only needed a very small *calming* of the initially imposed boundary conditions in order for the convergence to be regained.This explains the speed with which the boundary conditions may be introduced(in the second case)and still result in convergence.

Of course a drawback of using a gradual'run in'of boundary conditions(especially if they are extreme)is that the numerical solution is in fact solving a different

problem to that which would result if the boundary flow was introduced immediately.The gradual'run-in'is really only of use if the steady-state solution of the soil is sought.If the numerical solution must approximate the solution which introduces the flow immediately at *all* times then a much better approach is to initially reduce the time-step and then gradually increase it as the system recovers from the shock of the initially imposed boundary conditions.

3.4 Varying the time-step

As we indicated in the previous section,this seems(in most cases)to be the most superior way of letting the program deal with extreme or abruptly imposed boundary conditions.It is not difficult to modify the program to be able to cope with a varying time-step length.Some method of varying the time-step must be implemented so that the number of iterates achieved remains roughly constant(per time-step)over the running of the numerical solution.Of course the time-step length must only be varied slowly since a large variation may cause the time-stepping procedure to approximate a wildly inaccurate value of the next time-step and hence convergence may be threatened.However in our experimentations we found the program to be fairly robust to time-stepping changes,and easily able to cope with a halving of time-steps over one time-step.

We will now include a specific numerical example.We comment on a program which changes the time-step depending on the number of iterates being required to achieve convergence on the last time-step.The program is given a single number of iterates(or a range of iterate values)and if the program produces a number less than this(per time-step)then it increases the time-step by a factor of 10 per cent.If the program exceeds this number by 1 then the time-step is decreased by a factor of 10 per cent.If the program exceeds this number by a number greater than 1 then the time-step is halved.Hence usually the program achieves a constant number of iterates(per time-step).

Numerical example

In the following example we introduce a specified rainfall into the soil structure with specified initial pressure head.The depth of the soil sample is 40cm and the space-step is also specifed below.A tolerance of 10^{-3} cm was used(unless stated otherwise).A transfer term assuming rectangular shaped blocks of size 1cm was used.In all cases a'free-flow'boundary condition was imposed at the lower end of the soil structure.

We show the results in the form of a table 3.1 .In case (a) an initial time-step of 10^{-3} days and space-step of 1cm are used,with initial pressure heads of -100cm in both pores.We imposed'no-flow'boundary conditions at the top.We measured the total number of iterates to reach 0.1days.In case (b) we used the same initial conditions as in case(a) but now with an initial time-step of 5×10^{-4} days with initial pressure heads of -50cm in both pores.In this case a flow of 0.5cm/day was also imposed to both the matrix and fracture pores.We measured the total number of iterates required to reach 0.05days.In case(c) we used the same conditions as in

case(a) except the initial time-step was 10^{-4} days,the space-step was 0.25cm and We counted the total number of iterations up to time 0.005 days.In case(d)we used the same conditions as case(a)except now a tolerance of 10^{-2} cm was now used.Finally.in case(e) the same conditions were used,as in case(a),except now an initial pressure head of -50cm was imposed in both matrix and fracture pores.

We give a large range of cases to show that the behaviour observed is not just localised to a certain type of problem and initial conditions.

iterates per time step	total number of iterates for case(a)	total number of iterates for case(b)	total number of iterates for case(c)	total number of iterates for case(d)	total number of iterates for case(e)
2	320	450	107	86	472
3	214	1051	112	97	338
4	206	395	109	101	323
5	202	232	117	109	322
6	207	230	117	114	321
7	224	231	123	118	346
8	213	232	139	130	358
9	230	236	140	130	388
10	236	238	142	129	393
10-14	244	265	190	232	426

Table 3.1: Results

These results are somewhat erratic.This *randomness* can probably be accounted for by the inherent uncertainty resulting from the method of changing time-step.For example if the time-steps are such that the program often overshoots the desired number of iterates(per time-step)by more than 1 then the time-step is halved.Hence in some unlikely circumstances it is possible that asking the computer to increase the required number of iterates(per time-step)by 1 could actually result in a smaller average number of iterates(per time-step)actually resulting.Therefore if this method is to be adopted then a more reliable method of achieving a set number of iterates(per time-step)is probably required.

Nevertheless,our results still show that the optimum number of iterates to aim for(per time-step)is not a constant value(however it does not vary a great deal).Perhaps the best value to aim for is 5 because in these examples it is quite near the optimum point.

Assuming that we do not change time-step then the time-stepping estimation and the estimation made between iterates use $\Omega = 1.0$ and $\omega = 1.0$ (equations 2.24 and 2.25) respectively.Therefore it is difficult to see why there should be a benefit in aiming for 5 iterates per time-step because both the convergence of the iterates (see equation 2.24) and the time-stepping approximation(see equation 2.25)both relax with parameter 1.0.

By way of summary we can not emphasise enough the necessity of a varying time-step procedure.This is because at certain times the time-step must be very small because of extreme imposed boundary conditions or because the program seems to be tending towards a more saturated numerical solution.If the time-step

was not this small then the iteration of the solution(at a given time-step)would simply not converge to a solution for the next time-step in a finite number of iterations.However as the numerical solution changes(with time)the program may not need to maintain such a small time-step in order to maintain convergence(because conditions may have become less severe or may have become less saturated)and in some cases the time-step may safely increase by an order of more than 1000.Therefore unless a variable time-step is used,the program would be forced to have a very small time-step for all time,which results in an inefficient total number of iterations being required.A feasible method of time-step changing is one dependent on the number of iterates required for convergence at the last time-step(as this section shows).

Variable space-step may also have advantages,however it is much harder to vary the space-step(with time).Therefore we will not investigate that possibility.

3.5 Accelerating the convergence of the iteration

Numerical example

In this section we comment on the results which can be obtained by having an initially constant pressure head in both the fracture and matrix pores.Then'free flow'boundary conditions can be imposed at the bottom of the soil formation and a'no flow'boundary condition at the top of the soil structure.The exchange term was assumed to be zero for this problem(ie.each medium was assumed to be behaving in isolation).The space-step was set at 0.2cm and the time-step was set to 10^{-5} days (constant)(but these figures are largely irrelevant to this section and are just included for completeness).

The initial pressure heads and the relaxation parameter ω (as used in equation 2.24)were varied and the results are shown in Table 3.2. Table 3.2 shows the number of iterates required for convergence to a tolerance of 10^{-3} cm on the 4th time-step of the above problem for the different initial pressure head differences of -29cm and -20cm.

It can be seen from Table 3.2 that the optimum value of ω varies for initial pressure heads used.It is safe to assume that the optimum parameter probably also varies for completely different problems as well.

Therefore it is difficult to make any sweeping conclusions about which relaxation parameter is optimum for general convergence.However perhaps a safe conclusion to make is that overrelaxation is more likely to be of benefit than underrelaxation.This is an expected conclusion since the iterates do not oscillate.

In terms of general conclusion it also seems likely that too large a value of ω is more likely to lose convergence (and/or take a larger number of iterates to converge) than too small a value of ω.Therefore one may deduce that if one is unsure about which value to take one should always tend towards the smaller end of any value used,since it is better that a solution converges slowly than not

ω	number of iterates for initial pressure head in both pores of -20cm	number of iterates for initial pressure head in both pores of -29cm
0.6	13	
0.7	12	
0.8	11	
0.9	10	
1.0	10	4
1.1	9	4
1.2	9	3
1.3	8	3
1.4	8	3
1.5	8	4
1.6	7	5
1.7	7	
1.8	10	
1.9	16	

Table 3.2: Results

at all.This is especially true since the difference(as measured by the number of iterates required)produced by changing the value of ω is only relatively marginal.

Bearing the last two paragraphs in mind,we suggest that a fixed value of $\omega = 1.2$ may be the best value to use for a general problem.However,as the results show,particular cases can do much better.

On further investigation of the results we were interested in the following quantity(which we call $[\Psi]_j$)where

$$[\Psi]_j = |\frac{[\Lambda]_j}{[\Lambda]_{j-1}}| \tag{3.4}$$

where $[\Lambda]_j$ represents the maximum absolute difference between the pressure head values of the jth and (j+1)th iterate.In calculating values of Λ we were careful to take the pressure head value at the new(or (j+1)th)iterate as the value *before* the iterate was accelerated(using the current value of ω).

We found that when the initial pressure head value was -20cm the value of $[\Psi]_j$ settled at about 0.78 (after an initial fluctuation).However when the value of -29cm was used as the initial pressure head value we found that $[\Psi]_j$ was more in the region of 0.42.Therefore we conjecture that(assuming that the value of Ψ remains approximately constant over a time-step and that the iterations do not oscillate)the larger the value of Ψ obtained then the larger the optimal value of ω is.This could possibly form the basis of a method to accelerate convergence for any problem(in much the same way that the Cooley algorithem [Cooley 1983] can accelerate any problem which requires underrelaxation).

On further investigation we decided that the assumption of Ψ remaining constant over the time-step was not valid and so we obtained some further results which

ω	iter.(a2)	$[\Psi]_{2,2}$(a)	$[\Psi]_{4,2}$(a)	iter.(a3)	$[\Psi]_{4,3}$(a)	iter.(b2)	$[\Psi]_{4,2}$(b)
0.6	24	0.99051	0.999122	11	0.999862	12	0.999903
0.7	22	0.99050	0.998926	10	0.999851	11	0.999879
0.8	20	0.99048	0.998722	9	0.999840	10	0.999858
0.9	18	0.99048	0.998521	9	0.999831	9	0.999837
1.0	17	0.99046	0.998307	8	0.999830	9	0.999818
1.1	16	0.99045	0.998099	8	0.999833	8	0.999800
1.2	15	0.99043	0.997885	7	0.999842	7	0.999787
1.3	14	0.99042	0.997675	7	0.999857	7	0.999777
1.4	13	0.99040	0.997467	7	0.999880	7	0.999771
1.5	13	0.99039	0.997266	7	0.999917	9	0.999770
1.6	13	0.99035	0.997059	8	0.999974	12	0.999774
1.7	19	0.99030	0.996853	12	1.000047	18	0.999784
1.8	42	0.99028	0.996658	22	1.000051	41	0.999801
1.9	not conv.			not conv.			

Table 3.3: Results

we displayed in the following table(table 3.3).

In both cases a tolerance of 10^{-3} cm,average rectangular shaped matrix block size of 1cm and a space-step(fixed)of 0.2cm were taken.However, in addition, case(a)used a fixed time-step of 10^{-5} days with initial pressure heads initially set to -20cm in both media.Case(b)used a fixed time-step of 10^{-4} days and initially set both media to have -100cm pressure heads.The letter in brackets(within the table)refers to either of the two cases used.

The number of iterates was counted in order to achieve convergence.('iter.(a2)'refers to the number of iterates required for convergence on case(a)on the second time-step,and similiarly for the other labels).$[\Psi]_{4,2}$(a),$[\Psi]_{3,2}$(a),etc.simply refer to the value of $[\Psi]_4$,$[\Psi]_3$,etc.(as defined in equation 3.4)for case(a)on the 2nd time-step,and similarly for case(b).

As is clear(from case(b))there maybe some relationship taking place,since the optimum value of ω seems to be accompanied by the smaller values of Ψ.However any firm conclusions are shattered by some of the results from case(a).We conjecture that an acceleration scheme which trys to minimise Ψ (during the iteration of every time-step) would result in quicker convergence.Further investigation is needed.

We could argue that the total number of iterates for the time-step gives the best indication as to the efficiency of the iteration.However we only know this number *after* the time-step is complete and therefore it can not be used *during* the time-step to accelerate its convergence.However,by taking the value of Ψ before the time-step is complete,we hope that it may be possibe to vary ω during the iteration process to minimise Ψ and therefore hopefully accelerate the convergence *during* the iteration process.However,as the results show,much more work has to be done on this question.

Just for reference the Cooley algorithem[Cooley 1983] was implemented and since it repeatedly returned the maximum relaxation parameter ω =1.0 it is clear that

there is no benefit to be had from introducing it.Other methods of accelerating the iteration process (such as Newton[Conte-and-de-Boar-1980]) are also not applicable due to the effort required to form the various differentials.Newton also requires a very good starting value for convergence to be achieved.

However in [Gerke and van Genuchten 1993] Picard iteration with Cooley underrelaxation [Cooley 1983] is used which suggests that the iterations (of a time-step) *do* oscillate.This is because if their iterates *do not* oscillate then the underrelaxation scheme used would be of no benefit.It should be noted that although we suspect the iterates did oscillate this is **not** explicitly stated anywhere in [Gerke and van Genuchten 1993].

Since we suspect the iterates(of a time-step)oscillate when 'L1'mass lumping is used but the iterates do not oscillate when 'L2'mass lumping is used then we conclude that the 'L1' form of mass lumping is probably preferable.We conclude this because as long as the iterates(of a time-step)oscillate then their convergence may be accelerated by the Cooley algorithem.If the iterates do not oscillate then their convergence can not be easily accelerated.

The same procedure of varying the parameter value Ω as in equation 2.25 may be applied to the initial interate of the new time-step.However,since the solution does not oscillate in time either it seems likely that similar results will be obtained.We tried values of $\Omega = 0.5$ and $\Omega = 1.0$ and found the value of $\Omega = 1.0$ to be superior(as one would expect for a non oscillating problem).This value of Ω was then adopted.

3.6 Numerical difficulties

We now aim to quantify the numerical difficulty associated with solving different types of numerical problem.

Numerical example

To achieve this aim we set the computer a set of problems(or cases)which only vary by one aspect.To quantify the difficulty that the computer has we will produce a table similar to table 3.1 .However now only 5 iterates(per time-step)were aimed for and no other values were taken(ie.variable time-step was used as described in the previous section).

The larger the total number of iterates required,the harder the computer finds the problem(in general–there are some exceptions).There are different cases to be considered.

Case(a)is the same as case(a)from the previous section,ie.using a 40cm depth of soil,a tolerance of 10^{-3} cm,initial(variable) time-step of 10^{-3} days with initial pressure heads of -100cm in both matrix and fracture pores.We were assuming a transfer term which assumed that the average rectangular shaped matrix block size was 1cm and also setting the space-step to be 1cm.Finally a'no flow'condition was assumed on the upper surface and a'free flow'boundary condition was assumed on the lower surface.We counted the number of iterations required to

reach a time of 0.1 days.

In case(b)we used the same conditions as in case(a) except our initial(variable) time-step was 10^{-4} days,the space-step was 0.5cm.In case(o)a space-step of 2cm was used with other conditions remaining as those in case(a).The total numer of iterates required to solve each of these cases may be found in Table 3.4.

In case(c)we used the same conditions as case(a)except now a tolerance of 10^{-2} cm was used.The results of these cases are displayed in Table 3.5.

In case(d)the same conditions were used,as in case(a),except now an initial pressure head of -50cm was imposed in both matrix and fracture pores.In case(g)the same conditions were used as in case(a)except now that an initial pressure head of -500cm was used.Table 3.6 displays the results for these cases.

In case(e)the same conditions were used,except now a the average rectangular matrix block size was assummed to be 0.1cm.The conditions of case(h)were as those of(a)except now the average rectangular matrix block size was assumed to be 3.3cm.The results for these cases are displayed in Table 3.7.

In case(f)a soil depth of 80.0cm was assumed,with all other conditions to be left as in case(a).Also is case(n)the length of the soil structure was assumed to be 20cm with again the other conditions remaining as in case(a).Table 3.8 displays the results for these cases.

In case(i) the conditions were as those of(a)except now a flow of 0.2cm/day was imposed at the upper surface.The same conditions as case(i) are used in case(m) except now 0.1cm/day was applied to the upper surface.The results of these cases are displayed in Table 3.9.

In case(j)the matrix had initial conditions of a pressure head of -50cm and the fracture had initial conditions of a pressure head of -100cm with all other conditions being the same as those of case(a).In case(k)the conditions were those of case(j) but with the average matrix block size assumed to be 0.1cm.Finally in case(l)the same initial conditions were used as those in case(j) except now the average rectangular matrix block size was 3.3cm.Table 3.10 shows the results of these cases.

Our results show that there is an advantage to be had in increasing the tolerance of the iteration.Of course the price you pay for this advantage is a reduction in the accuracy of the solution.From the results from case(a) and case(c)(Table 3.5)we can say that there is a significant difference in the results.It is safe to assume that the results of case(a) will be substantially more accurate than those of case(c).Where case(a)has a pressure head of -107.1cm,case(c)has a pressure head of -108.9cm(at the same depth),however in both cases the qualitative nature of the numerical results are similar.Furthermore this error would probably be acceptable,from a hydrological point of view,because of the many uncertainties which are associated with the various parameters used for *real* problems.The absolute error(between results)which we have experienced here would be expected to increase with time(for similar time-steps).

total number of iterates for case(a)	total number of iterates for case(b)	total number of iterates for case(o)
202	528	82

Table 3.4: Table showing the effects of varying space-step size

total number of iterates for case(a)	total number of iterates for case(c)
202	109

Table 3.5: Table showing the effects of varying tolerance

total number of iterates for case(a)	total number of iterates for case(d)	total number of iterates for case(g)
202	322	55

Table 3.6: Table showing the effects of varying initial pressure heads

total number of iterates for case(a)	total number of iterates for case(e)	total number of iterates for case(h)
202	193	204

Table 3.7: Table showing the effects of varying matrix block sizes for *equal* pressure heads in fracture and matrix pores

total number of iterates for case(a)	total number of iterates for case(f)	total number of iterates for case(n)
202	202	202

Table 3.8: Table showing the effects of differing soil depth

total number of iterates for case(a)	total number of iterates for case(i)	total number of iterates for case(m)
202	402	188

Table 3.9: Table showing the effects of applying rainfall

total number of iterates for case(j)	total number of iterates for case(k)	total number of iterates for case(l)
354	294	317

Table 3.10: Table showing the effects of varying matrix block sizes for *differing* pressure heads in fracture and matrix pores

Our results also show that there is a very large increase in the amount of numerical work required to find a solution with a halved space-step.In fact this increase in work is even larger than is suggested by the table (see table 3.4)because the amount of numerical work done(per iterate)was larger in case(b)than case(a)because there were a greater number of spatial nodes present.

As we have suggested earlier on in the dissertation,it seems(from the results of table 3.6)that the program computes less saturated conditions more easily than more saturated conditions.

The results also show us that changing the average matrix block size seems to have a relatively small effect on the increase in the amount of numerical work required when the pressure heads in the two media are similar(see Table 3.7).However the effects seem to be more marked when the pressure head differences are larger(see Table 3.10).

It also seems that doubling or halving the soil depth does not change the total number of iterates required(see Table 3.8).However,as with Table 3.4,the amount of numerical work required is still substantially increased.

When some rainfall is applied(0.2cm/day)there is a significant increase in the amount of numerical work needed to obtain a numerical solution(see Table 3.9).This is probably as a result of the'shock'of immediately introducing the boundary conditions and not so much as a result of the numerical solution becoming more saturated(as it does not have time to become significantly saturated).In fact applying a slightly lesser amount of rainfall(eg.0.1cm/day as case(m)shows)can actually need less numerical work than case(a) required, which had no rainfall applied.We think the reason for this is because 'no-flow'boundary conditions at the top of the soil structure actually cause more of a 'shock' to the program than just letting a small amount of water flow into the surface to replace the water 'free-flowing' out of the soil at the lower end.

3.7 Evaluating the functions at mid-points

As is mentioned earlier in this dissertation(in Chapter 2),the way in which the value of the functions (ie the hydraulic conductivity of the specific soil water capacity) are taken at the mid-point of two nodes is to evaluate the functions at the pressure head values of the two adjacent nodes and then take the arithmetic average.

However another way of attacking the problem is to average the pressure head val-

ues at the two adjacent nodes *before* the function's value is calulated(ie.evaluate the functions' value at the averaged pressure head).In so doing one will find that there is a significant reduction of the number of function calls required.Since the function evaluations are somewhat complicated and therefore time consuming,there is a significant advantage to be had in averaging the pressure head values before calling the function.

Numerical example 1

We set a tolerance of 10^{-3} days, initial pressure heads of -100cm in both pores, initial(variable) time-step of 10^{-3} days, space-step to be 1cm, depth of soil to be 40cm, average rectangular matrix block size to be 1cm, the number of iterates aimed for(with variable time-step)was 5, and finally we take the results at time = 0.2 days.

Using the averaging *after* the function calls the computer took 34.9 seconds(in *real* time)and used 359 iterates.In using the averaging *before* the function calls the computer took 19.7 second and 358 iterates.(Please note that we are convinced that the differences in *real* time that we observed are not as the result of variable demand on the computers,since we observed these differences several times)

The numerical results were remarkabley similar.For a pressure head value of -110.60427cm(at a particular depth)on the averaging *before* method, a pressure head of -110.60582cm was experienced on the averaging *after* method.This difference could safely be said to be neglible.

Since the averaging *before* method evaluates the function at the mid-point,and the averaging *after* method takes an arithmetic average of the values at the two node points it is clear that any discrepancy between the two methods will be greater when the space-step and/or time-step is larger(for a constant soil depth).

Numerical example 2

We now put this hypothesis to the test, by using the same conditions as the above numerical example-except we now use a space-step of 10cm(however a complication is that the average time-step may now be smaller than in Numerical example 1).

Averaging *after* took 66 iterates and 1.0 seconds and averaging *before* took the same numer of iterates and approximately half the time(although the time measurement was difficult to evaluate for such a short space of time).

Even now there is a very small difference in the numerical results.For example averaging before obtained a value of -106.344577cm (at a certain depth) and averaging after obtained a value of -106.365380cm (at the same depth).

We now aim to quantify the effect of a larger time-step in the following numerical example.

Numerical example 3

In the following numerical example we follow the conditions of Numerical Example 1(of this section),except now we let the program run for a substantially longer period of time(10.0 days).This means the time-step grows larger because(as the last section shows)the computer finds more ease in evaluating conditions which become more dry and tend more towards the steady-state solution.We also have a larger space-step(10cm)in order to exaggerate any differences.

Again now very little difference between the results occurs.Both versions take 416 iterates.By way of difference in numerical solution we can say the averaging *before* method has a value of -171.0261cm (at one depth)and the averaging *after* method has a value of -170.7046cm (at the same depth).Of course(as with all of these numerical examples)the qualitative nature of the results were also similar in the two cases.

Which is the correct method is a matter of debate.There should be little difference between the two approaches if the pressure head values and the functions called (ie K and c) vary linearly between the node points.However if they do not vary linearly then there will be some difference. For small space and time-step there should be little difference(as we have discovered).

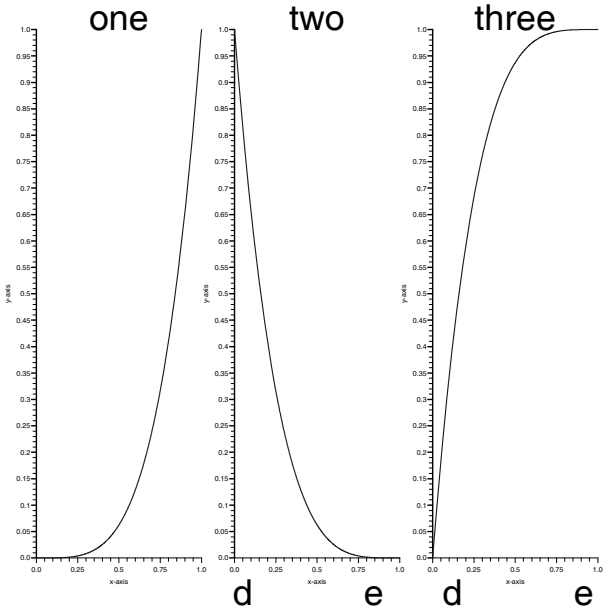

Figure 3.2: A possible distributions of nonlinear functions.

It is too complicated to analyse the case for general nonlinear functions,however we will come to a conclusion as to which is the best approach to adopt in some cases of special nonlinear functions.

From the next chapter (see figures 4.1 and 4.2) it will become clear that the distribution of the function K (the hydraulic conductivity function) is like that shown in Fig.3.2 case one,with the value of K being the y-axis and the pressure head being the x-axis.The similarity will not be exact, but hopefully it will be close enough so that the qualitative arguments we comment upon here follow through.

For the purpose of simplicity we will assume that the pressure head values may take either Fig.3.2 case two or Fig.3.2 case three(with the y-axis being the value of the pressure-head and the x-axis being the depth of the soil structure).This is a large assumption which is motivated by the form of the boundary conditions at the lower surface of the soil(these two types of pressure head distributions can be seen in the numerical results of figures 4.6 , 4.7 , 4.9 and 4.10).This assumption will be especially wrong when oscillatory boundary conditions are to be applied to the soil.

We will now explore the accuracy of the two methods of evaluation of the value of the K function at a mid-node in *space*.Firstly we will assume that the pressure head distribution has the form of Fig.3.2 case two.

Here the average value of the pressure heads at the two nodes(d and e in case 2 of figure 3.2)will be 0.5.By observing the assumed distribution for K(case one) it is clear that h=0.5 returns a value of about 0.05.For h=1(the value of the pressure head at node d)we find that K returns a value of 1.0 and for h=0(the value of the pressure head at node e)we find that K returns a value of 0.0.Therefore by averaging *after* we obtain a value of 0.5 for the value of the function K at the midpoint of the two nodes.

In fact the value of the pressure head(case two) at depth=0.5(ie.the midpoint of the two nodes,d and e,in space)is 0.05,which returns a value of K(case one) being approximately equal to 0.0.So in *this* (ie.case two) averaging *before* is the more accurate of the two methods and it will also require less computational time to evaluate the solution.

However a similar argument concludes that if the pressure head distribution has the distribution as given in case three then the averaging *after* scheme is the more accurate(or the closer to the correct value).

Therefore we conclude that,for these two simple examples,neither of the two methods is superior on a general pressure head distribution.This implies that it is better to use the averaging *before* method since it does not lose out on accuracy and produces a result in about half the time.

As stated earlier,we have used a large number of simplifying assumptions.We will comment on the inadequacies of our previous argument as follows.

Firstly we have not discussed at all how the pressure head values may vary with time.This is significant because a large amount of timelike averaging between nodes is also necessary.However we hope the distribution may be similar to that of case two or case three,but have no results to prove this.

Secondly the 'c' (ie.the specific soil water capacity) functions are also averaged between time nodes and again we have no results(as yet)to give us any indication as to the distribution of this function(in time).However (for non-oscillatory solutions) perhaps we should hope that the distribution is quite close to case two or case three.

Finally(as we have mentioned before)in certain circumstances the spatial distri-

bution of the pressure head values may not be similar to either case two or case three, and hence all of the preceeding arguments may be invalid.

Which method is best is open to question,however in any case(due to small space and time-steps)the differences between them appears to be very small anyway.Therefore we suggest the averaging *before* method is better to use on the grounds of the lesser amount of numerical effort required.

Chapter 4

Numerical examples and representation of functions used

4.1 Representation of functions used

The functions we have used in this model vary with pressure head values.Therefore it is useful to include their representation in my dissertation.

It is important to remember that these are just mathematical approximations to what occurs physically.They are not exactly correct and other approximations to the physical behaviour of the soil also exist which are not necessarily the same as the ones we are using here.

The first function we will consider is the hydraulic conductivity function $K_{m/f}$ (as introduced in chapter 1).These values are especially valuable to know because they can be used in conjunction with equation 1.7 and 1.8 to estimate the flux of fluid flowing.

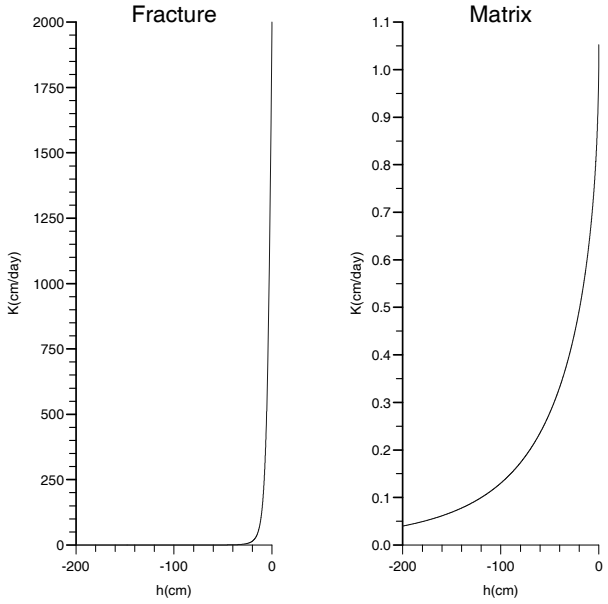

Figure 4.1: Hydraulic Conductivity values.

It is also helpful to compare the hydraulic conductivity functions on the same

graph also.This is shown in figure 4.2

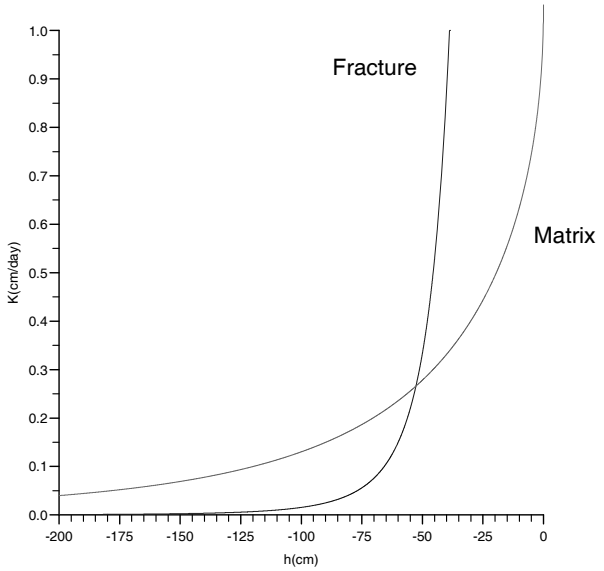

Figure 4.2: Hydraulic Conductivity values.

The water content functions ($\theta_{f/m}$ as introduced in chapter 1) also vary as a function of pressure head and are shown(as a proportion of saturated value)in the following graph(ie.figure 4.3).

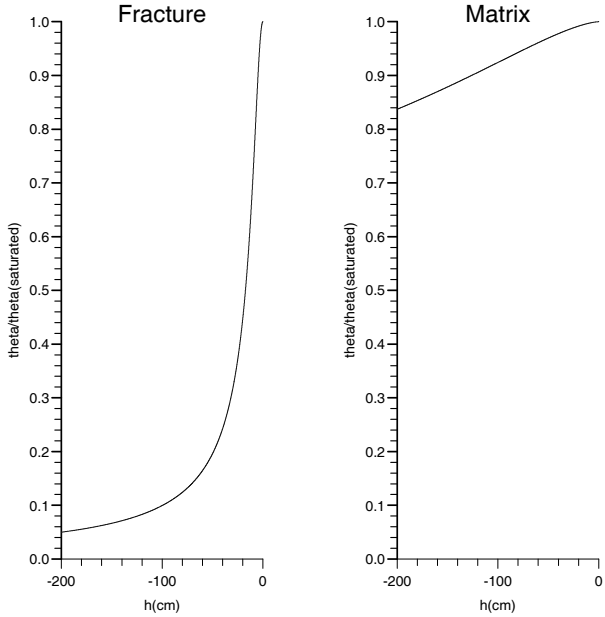

Figure 4.3: Water content functions.

Finally the $c_{m/f}$ function (as introduced in chapter 1)is also a variable function of pressure head.Its value is given by equations 1.5 and 1.6 .Its representation is given in figure 4.4 .

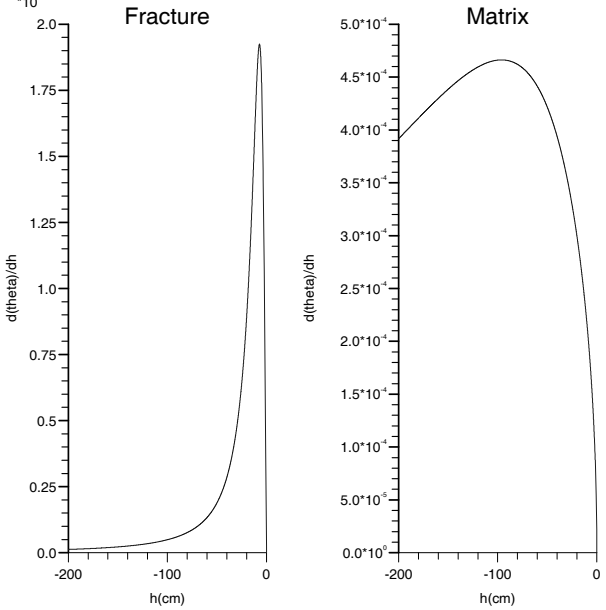

Figure 4.4: The specific soil water capacities.

4.2 Numerical examples

In this section we aim to show that the model and the numerical solution at least give plausible results and that,therefore,the dual porosity model is worthy of further investigation.Since there are many factors and variables to consider when deciding if the numerical solution is sensible,we will try and concentrate on simple numerical examples so we will have a better chance of understanding the processes taking place better.

4.2.1 Very Wet Soil

Numerical Example

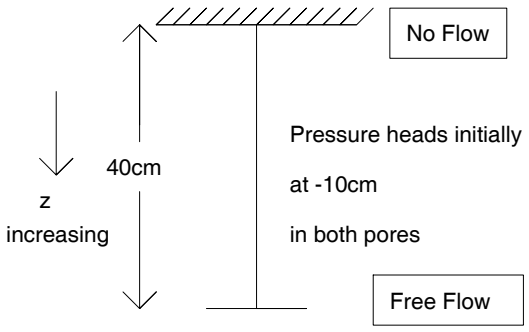

Figure 4.5: The numerical example.

In this numerical example we will show how the model reacts to an initially wet soil(see figure 4.5).This soil will slowly dry because we have a'no flow'boundary condition at the top of the soil structure and a 'free flow'boundary condition at the bottom of the soil structure.The initial conditions were of a pressure head of -10cm in both fracture and matrix pores.The length of the soil medium was 40cm, space-steps of 1cm were used and the average rectangular matrix block size was set to be 1.0cm.

By observation of figure 4.3 it is clear that just because the pressure heads are identical(initially)in the matrix and fracture pores does not necessarily mean that their water contents are the same.Nevertheless it is still clear that both media would initially be said to be'wet'as defined by any sensible definition(in fact a more precise definition of 'wetness' may be given by equations 1.11 and 1.12).Further,since both the water content functions in the matrix and fracture are monotonic,it is a safe to assume that as the pressure head decreases in one medium then the water content of that medium also decreases.

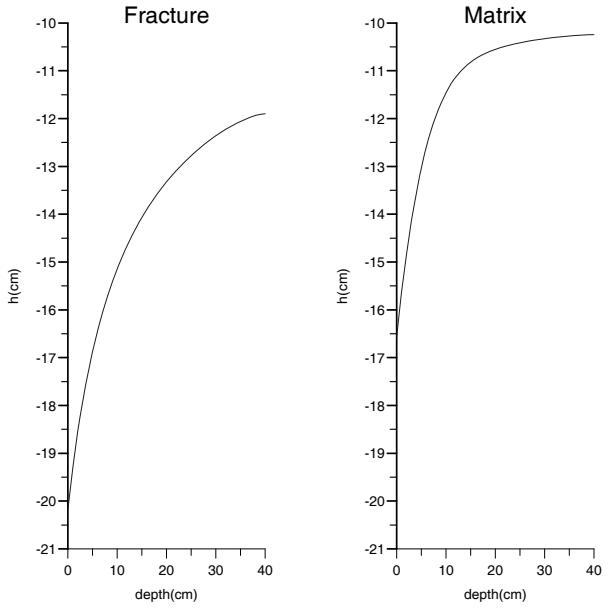

Figure 4.6: Pressure Heads at time=0.01 days

By observing figure 4.6 (also with reference to figure 4.3)it is clear that the pressure heads in the fracture are decreasing faster than the matrix.This is because the water is flowing downwards more quickly in the fractures.The fact that the water is flowing quicker in the fractures can be deduced from figure 4.2 and from equations 1.7 and 1.8 ,given the initial pressure heads used(ie.initially zero pressure head gradients).

This phenomenon may be thought of as the water running quickly down the wet cracks while the water seeps only slowly down the wet matrix blocks which seems to be intuitively correct.This process would not be modelled as well by a single porosity model.

It is also clear that the water has had little effect on the lower boundary of the matrix pores since the pressure head values are still around the values to which it was

set initially.This may be accounted for by the explanation that since the water is flowing(relatively)slowly in the matrix pores,it has had not had time to make the lower end of the soil structure significantly dryer at this short time(0.01days)after the initial conditions were imposed.

At this point we should perhaps mention the transfer term.This regulates the transfer of water from the medium with the higher pressure head to the medium which has the lower pressure head(its definition is given by equations 2.3 and 2.26).We will assume that this transfer term has had little effect on the flows observed in figure 4.6 since it will have not had time to affect the flow significantly because the pressure head differences are not of a large enough magnitude.

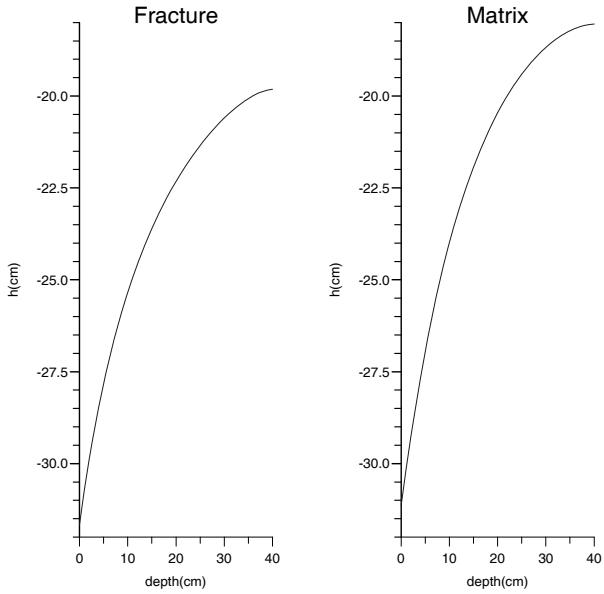

Figure 4.7: Pressure Heads at time=0.1 days

Now considering figure 4.7 (again also with reference to figure 4.3)it is clear that the fracture system has dried significantly more than at time 0.01days and the matrix is also dryer than it was at time 0.01days.This is(at least partly)because(as commented upon before)the water travels faster(at first)in the fractures than in the matrix.

By again considering figure 4.7 further we are beginning to see a different phenomenon taking place.Since initially the flow is greater in the fracture than the matrix it seems likely that the majority of the dryness of the matrix pores may be accounted for by the transfer term(γ_w as introduced in equations 2.3 and 2.26)moving water into the fracture.We think this process is now significant because of the time the process has been running ,the differences in the pressure head values experienced at t=0.01 days and the growing similarity between the two pressure head distributions of the two media at 0.1days.

It may be possible to analyse the hydrology of the processes occurring in the numerical results further(by calculating the flux of water flowing in the two pores-for example)but that degree of hydrological detail has not be carried out because of time constraints.

By imagining what would happen physically in the soil and by analysing the numerical results in this example it becomes obvious that this dual-porosity model gives very believable results and hopefully models the movement of unsaturated flow in the soil much better than the previously used single porosity models(see [Beven and Germann 1982]for more information on the inadequacy of single porosity models).

4.2.2 Applying Rainfall

The average size of the rectangular matrix blocks is a critical factor affecting the transfer of water between the two pores.This is adequately discussed in the 1993 paper of Gerke and van Genuchten.However we will just include two examples to show how well the dual porosity model deals with differing average rectangular matrix block sizes and to show how well the model deals with rainfall.

It should be noted that some care is needed in devising the problem because if a much dryer initial condition is used , or if a much larger flux of rainfall is applied then we risk forming a positive pressure head at the top of the soil structure(usually at the fictitious point).This is probably an undesirable situation and although we can cope with this in certain circumstances(see section 1.2), we do not know how to overcome this problem in general.Therefore we will only use situations which are not so severe as to cause a positive head to be formed at the fictitious point.

Numerical Example 1

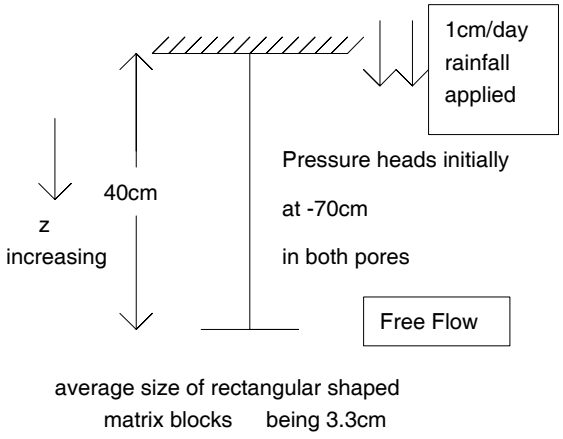

Figure 4.8: The numerical example.

Again we use a 40cm depth of soil with 1cm space-steps.We will immediately apply rainfall of 1cm/day to an initially dry (-70cm of pressure heads in both soils) soil to see how the model reacts to this situation.Figure 4.8 shows the various boundary and initial conditions applied.

Here it is clear(by observing figure 4.9)that the water flux,due to rainfall,has

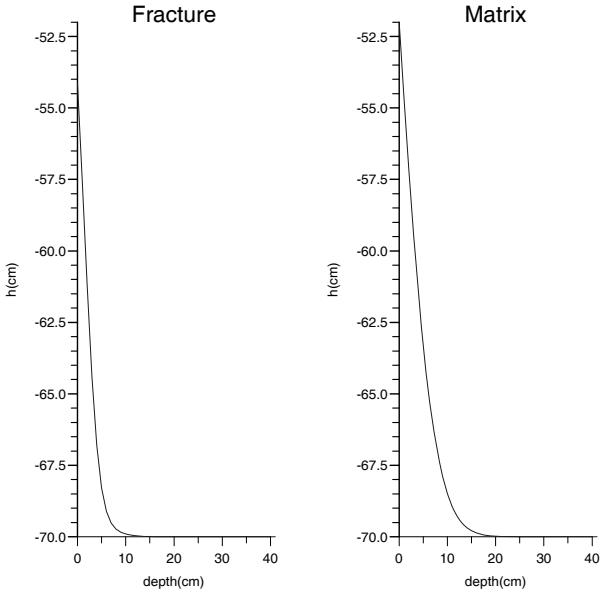

Figure 4.9: Pressure Heads at time=0.04 days with average rectangular matrix block size=3.3cm

infiltrated further into the matrix than fracture pores at this time.This is what one may expect given the initial pressure heads,by observing figure 4.2 and using equations 1.7 and 1.8 .

Since the average matrix block size is relatively large(and therefore the transfer term(equations 2.3 and 2.26)is relatively small)and a relatively short time has been allowed to pass since the imposition of the boundary conditions it is safe to assume that there has been relatively little transfer of water between the two pore systems.

Numerical Example 2

We will now show an example in which a smaller average rectangular matrix block size(1.0cm)is used to show that now the pressure heads become more similar(all other initial and boundary conditions are as in Numerical Example 1 of this subsection).

By observing figure 4.10 it is clear that there has been a larger transfer of water occurring between the two systems.Physically this is explained because the smaller rectangular shaped matrix blocks have more surface area from which to pass water.

One must also bear in mind the larger volume of matrix pores as opposed to fracture pores (as dictated by the w_f term as given in table 2.1 and as introducted in section 1.2.) when trying to analyse these results.That means that when the water transfer is increased between the two pore systems then the pressure head values in the fracture are likely to change more than the pressure head values in the matrix pores.

Again the dual-porosity model has proved itself to be superior to the single-porosity models available,since the single porosity models would have a severe

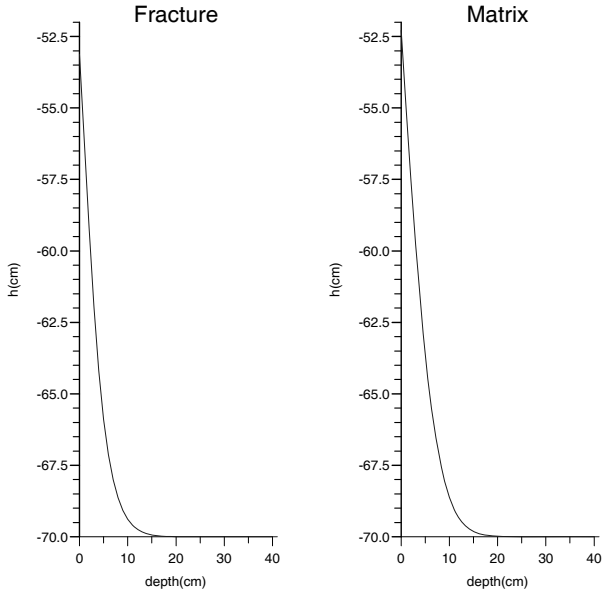

Figure 4.10: Pressure Heads at time=0.04 days with average matrix size=1.0cm

difficulty in modeling the interaction between the two pore systems which is resulting here and which (of course) will be dependent on the average matrix block size as figures 4.9 and 4.10 show.

4.3 Conclusion

No oscillation was observed in time,space or in the consecutive iterates of a time-step.

The gradual introduction of rainfall was found to have a beneficial effect on the convergence of the numerical solution(as opposed to its immediate introduction)

A variable time-step was found to be a necessity for producing our numerical solution in an efficient number of time-steps.

The convergence of the iteration(of a time-step)may be accelerated by overrelaxation.However it is not clear how this acceleration may be maximised in general

In general dryer(ie.problems with more negative pressure head values)are solved with less computational work being required.Also problems which have smaller depth,larger space-step and larger tolerance are computed with more computational ease.

The difference in the amount of computational work required when changing the average size of the rectangular shaped matrix blocks is marginal when the pressure head differences between the two pores systems is small.However when the pressure head differences between the two pore systems is larger then the difference in the computational work is more marked.

The imposition of increased rainfall at the upper surface increases the computational effort required,in general.

We also conclude that it is probably better to calculate nonlinear terms by averaging the values of the pressure heads to adjacent nodes(in time or space)*before* calulating the corresponding hydraulic conductivity or specific soil water capacity.

Finally(as the graphical examples show)we conclude that the dual-porosity model produces plausible results for certain initial and boundary conditions.

Bibliography

[Beven and Germann 1982] Keith Beven and Peter Germann.*Macropores and Water Flow in Soils.*Water Resources Research,Vol.18,No.5,Pages 1311-1325,(October 1982).

[Conte-and-de-Boar-1980] Conte ,S.D. and de Boor ,C.*Elementary numerical analysis.*McGraw-Hill,(1980).

[Cooley 1983] Cooley ,R.L.*Some new procedure for numerical solution of variably saturated flow problems.*Water Resources Research,19,1271-1285,(1983).

[Darcy 1856] Darcy ,H. *Les fontaines publiques de la ville de Dijon.*Dalmont,Paris,(1856).

[Gerke and van Genuchten 1993] Gerke ,H.H. and Van Genuchten ,M.T.*A Dual-Porosity Model for simulating the Preferential Movement of Water and Solutes in Structured Porous Media.*Water Resources Research,29,305-319,(1993).

[Milly 1985] Milly,P.C.D.*A mass-conservative procedure for time-stepping in models of unsaturated flow.*Adv.Water Resources,Volume 8,32-36,(March 1985).

[Neuman 1973] Neuman ,S.P.*Saturated-unsaturated seepage by finite elements.*J.Hydraul.Div.Am.Soc.Civ.Eng.,99,2233-2250,(1973).

[Ouyang and Xiao 1994] Huajiang Ouyang and Ding Xiao.*Criteria for elimination oscillation in analysis of heat-conduction equation by finite-elent method.*Commumications in numerical methods in engineering,Vol.10 453-460,(1994).

[Richards 1931] Richards ,L.A.*Capillary conduction of liquids through porous mediums.*Physics,1,318-33,(1931).

[van Genuchten 1980] van Genuchten ,M.T.*A closed-form equation for predicting the hydraulic conductivity of unsaturated soils.*Soil Sci.Soc.Am.J.,44,892-898,(1980).

[Wood 1993] Wood ,W.L.*Introduction to numerical methods for water resources.*Clarendon Press,Oxford,(1993).

[Wood and Calver 1990] Wood ,W.L.and Calver ,A.*Lumped Versus Distributed Mass Matrices in the Finite Element Solution of Subsurface Flow.*Water Resources Research,26,819-825,(May 1990).

Symbol.	Description.
a(L)	average distance from centre of matrix block to its bounday.
$c_m(L^{-1})$	specific soil water capacity of matrix pores.
$c_f(L^{-1})$	specific soil water capacity of fracture pores.
$\underline{h}(L)$	average pressure head of matrix and fracture pores.
$h_f(L)$	pressure head of fracture pores.
$h_m(L)$	pressure head of matrix pores.
$K_a(L/T)$	hydraulic conductivity of fracture/matrix interface.
$K_f(L/T)$	hydraulic conductivity of fracture pores.
$K_m(L/T)$	hydraulic conductivity of matrix pores.
$K_{sf}(L/T)$	hydraulic conductivity at saturation of fracture pores.
$K_{sm}(L/T)$	hydraulic conductivity at saturation of matrix pores.
m_m	$m_m = 1 - \frac{1}{n_m}$
m_n	$m_n = 1 - \frac{1}{n_n}$
n_m	experimental constant for matrix pores.
n_f	experimental constant for fracture pores.
l	experimental constant
$q_f(L/T)$	flux of water in fracture pores
$q_m(L/T)$	flux of water in matrix pores.
s_{ef}	effective saturation of fracture pores
s_{em}	effective saturation of matrix pores
t(T)	temporal variable(starting with t=0 initially)
w_f	volume of fractual pores as a proportion of total volume
w_m	volume of matrix pores as a proportion of total volume
z(L)	spatial variable(measured downwards with z=0 being ground-level).
$\alpha(L^2/T)$	constant.
$\alpha_w^*(L^{-2})$	experimental constant depending on size and structure of matrix pores.
β	empirical constant depending on the structure of the soil.
γ	empirical constant.
$\gamma_w(T^{-1})$	term representing water transfer from fracture to matrix pores
$\Delta t(T)$	time-step.
$\Delta t_k(T)$	time stepped by the kth time step.
$\Delta z(L)$	spatial-step.
θ	parameter in the θ method of time stepping.
θ_f	soil water retention function for fracture pores.
θ_m	soil water retention function for matrix pores.
θ_{rf}	residual soil water retention constant for fracture pores.
θ_{rm}	residual soil water retention constant for matrix pores.
θ_{sf}	saturated soil water retention constant for fracture pores.
θ_{sm}	saturated soil water retention constant for matrix pores.
Λ	maximum absolute difference in pressure heads of consecutative iterates.
Ψ	specified ratio of Λ terms as given in equation 3.4.
ω	relaxation parameter in the iteration of a time step.
Ω	relaxation parameter in the time stepping approximation.