Richards'equation 1D

29 settembre 2016

1 Problema 1D

Nel caso unidimensionale l'equazione di Richards si scrive nella forma:

$$\frac{\partial \theta(\psi)}{\partial t} = \frac{\partial}{\partial x}\left[K(\psi)\frac{\partial \psi}{\partial x} + K(\psi)\right] \tag{1}$$

dove il legame tra il contenuto d'acqua del terreno e la suzione è espresso tramite la formula di Van Genuthgen:

$$\theta(\psi) = \begin{cases} \theta_r + \dfrac{\theta_s - \theta_r}{(1 + |\alpha\psi|^n)^m} & \psi < 0 \\ \theta_s & \psi \geq 0 \end{cases} \tag{2}$$

il legame tra la conducibilità idraulica e la suzione è espresso come:

$$K(\psi) = K_s \sqrt{\frac{\theta(\psi) - \theta_r}{\theta_s - \theta_r}}\left[1 - \left[1 - \left(\frac{\theta(\psi) - \theta_r}{\theta_s - \theta_r}\right)^{1/m}\right]^m\right]^2 \tag{3}$$

Per risolvere il problema è necessario fornire:

- una condizione iniziale su tutto il dominio, si assume una distribuzione idrostatica della pressione;
- due condizioni al contorno. In questo problema si impone il valore della suzione nel tempo al bordo del dominio (Dirichlet).

2 Schema numerico

Per risolvere l'equazione di Richards 1D si adotta lo schema BTCS implicito:

$$\theta_i^{n+1} = \theta_i^n + \frac{\Delta t}{\Delta x}\left[K_{i+1/2}^n \frac{\psi_{i+1}^{n+1} - \psi_i^{n+1}}{\Delta x} + K_{i+1/2}^n - K_{i-1/2}^n \frac{\psi_i^{n+1} - \psi_{i-1}^{n+1}}{\Delta x} - K_{i-1/2}^n\right] \tag{4}$$

dove:

- Δt è lo step temporale;
- Δx è la dimensione della cella;
- i indica il numero della cella;
- n indica lo step temporale;
- $\theta_i^n = \theta(\psi_i^n)$, è il contenuto d'acqua nella cella i-esima allo step temporale n-esimo;
- $K_i^n = K(\psi_i^n)$, è la conducibilità idraulica nella cella i-esima allo step temporale n-esimo
- $K_{i+1/2}^n = \dfrac{K_i^n + K_{i+1}^n}{2}$, è il valore della conducibilità all'interfaccia di due celle adiacenti.

Scrivendo l'equazione 4 per ciascuna cella si ottiene quindi un sistema non lineare che scritto in forma matriciale è:

$$\boldsymbol{\theta} + \boldsymbol{T}\boldsymbol{\psi} = \boldsymbol{rhs} \tag{5}$$

dove:

- $\boldsymbol{\theta}$ è il vettore che contiene i valori del contenuto idrico per ciascuna cella del dominio.
- $\boldsymbol{T}$ è la matrice dei flussi, è una matrice tri-diagonale del tipo:

$$\begin{bmatrix} b & c & 0 & 0 & 0 \\ a & b & c & 0 & 0 \\ 0 & a & b & c & 0 \\ 0 & 0 & a & b & c \\ 0 & 0 & 0 & a & b \end{bmatrix} \tag{6}$$

  che costruisco tramite i vettori $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$.
  Per risolvere il sistema posso usare il metodo di Thomas.

- $\boldsymbol{\psi}$ è il vettore delle incognite.
- $\boldsymbol{rhs}$ è il vettore dei termini noti.

3 Matlab

3.1 Funzioni

Le funzioni implementate sono:

- *kappa*: calcola il valore della conducibilità idraulica a partire dal valore della suzione. Come relazione di chiusura per la conducibilità idraulica si usa l'equazione 3.

  ```
  function K=kappa(psi)
  global alpha thetas thetar n m Ks psic
  sat = (Thetaf(psi)-thetar)/(thetas-thetar); %saturation is between 0 1
  K = Ks*sqrt(sat)*(1-(1-sat^(1/m))^m)^2;
  ```

  Alla funzione *kappa* si passa il valore della suzione.

- *Thetaf*: calcola il valore del contenuto d'acqua all'interno della cella a partire dal valore della suzione. Come relazione di chiusura per il contenuto d'acqua di usa l'equazione 2

  ```
  function y=Thetaf(psi)
  global alpha thetas thetar n m Ks psic
  if (psi<=0)
      y = thetar + (thetas-thetar)/( (1+abs(alpha*psi)^n)^m );
  else
      y = thetas;
  end
  ```

- *Theta1*: calcola il valore della variabile θ_1 come:

  ```
  function y = Theta1(psi)
  global alpha thetas thetar n m Ks psic
  if (psi<=psic)
      y = Thetaf(psi);
  else
      y = Thetaf(psic) + dTheta(psic)*(psi-psic);
  end
  ```

  (CasulliZanolli formula 9)

- *Theta2*: calcola il valore della variabile θ_2 come:

```
function y = Theta2(psi)
global alpha thetas thetar n m Ks psic
y = Theta1(psi) - Thetaf(psi);
```

  (CasulliZanolli formula 9)

- *dTheta2*: calcola il valore di $q(\psi)$:

```
function y = Theta2(psi)
global alpha thetas thetar n m Ks psic
y = Theta1(psi) - Thetaf(psi);
```

  (CasulliZanolli formula 7)

- *dTheta1*: calcola il valore di $p(\psi)$

```
function y=dTheta1(psi)
global alpha thetas thetar n m Ks psic
if (psi<=psic)
    %left of critical value, take the original derivative
    y = dTheta(psi);
else
    %on the right of the critical value, keep the maximum derivative
    y = dTheta(psic);
end
```

  (Casulli formula 7)

- *dTheta*: calcola il valore della capacità idraulica del suolo:

```
function y=dTheta(psi)
global alpha thetas thetar n m Ks psic
if (psi<=0)
    %unsaturated medium
    y = alpha*n*m*(thetas-thetar)/((1+abs(alpha*psi)^n)^(m+1))*abs(a
else
    %saturated medium
    y = 0;
end
```

3.2 BTCS

Le variabili $alpha$, $thetas$, $thetar$, n, m, Ks, $psic$ sono definite come variabili globali.

- $alpha$, n, m sono i parametri dell'equazione di Van Genuthgen;
- $thetas$ contenuto d'acqua a saturazione;
- $thetar$ contenuto d'acqua residuo;
- Ks conducibilità a saturazione;
- $psic$ è il valore della suzione per il quale si ha il massimo valore della capacità idraulica del suolo (è il valore $\psi*$ nell'articolo CasulliZanolli).

La condizione iniziale si assume una distribuzione idrostatica delle pressioni:

```
for i=1:IMAX
    psi(i) = -x(i);
end
```

All'interno del ciclo nel tempo vengono definite le condizioni al contorno:

- condizione al contorno di destra è funzione del tempo:

```
if(time<=1e5)
        psiR = -0.05+0.03*sin(2*pi*time/1e5);
    elseif(time>1e5 && time<=1.8e5)
        psiR = +0.1;
    else
        psiR = -0.05+2952.45*exp(-time/18204.8);
    end
```

- condizione al contorno di sinistra costante nel tempo:

```
psiL = 0;
```

si calcolano quindi le conducibilità idrauliche al contorno

```
KR = kappa(psiR);
KL = kappa(psiL);
```

Per ciascuna cella si calcola il contenuto d'acqua e la conducibilità idraulica con il valore della suzione calcolato allo step precende (eccetto al primo step che si parte dalla condizione iniziale).

```
for i=1:IMAX
    theta(i)=Thetaf(psi(i));
    K(i) = kappa(psi(i));
end
```

Si calcola per ciascuna cella il termine noto, $\boldsymbol{rhs}$, e le componenti dei vettori $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$ che formano la matrice dei flussi, $\boldsymbol{T}$.

Per la cella $i = 1$ l'equazione 4 porge

$$\theta_1^{n+1} = \theta_1^n + \frac{\Delta t}{\Delta x}\left[K_p^n \frac{\psi_2^{n+1} - \psi_1^{n+1}}{\Delta x} + K_p^n - K_m^n \frac{\psi_1^{n+1} - \psi_L^{n+1}}{\Delta x/2} - K_m^n\right] \tag{7}$$

dove:

- ψ_L e K_L indicano rispettivamente la suzione e la conducibilità idraulica in corrispondenza del bordo sinistro del dominio;
- K_p è la conducibilità idraulica all'interfaccia tra le celle $i = 1$ e $i = 2$;
- K_m è la conducibilità idraulica all'interfaccia tra le celle $i = 1$ e $i = 0$;

Riscrivendo l'equazione 7 portando a destra i termini noti e a sinistra le incognite si ottiene:

$$\theta_1^{n+1} - \frac{\Delta t}{\Delta x}\left[K_p^n \frac{\psi_2^{n+1}}{\Delta x} - K_p^n \frac{\psi_1^{n+1}}{\Delta x} - 2K_m^n \frac{\psi_1^{n+1}}{\Delta x}\right] =$$

$$\theta_1^n + \frac{\Delta t}{\Delta x}\left(K_p^n - K_m^n\right) + \frac{\Delta t}{\Delta x} 2K_m^n \frac{\psi_L^{n+1}}{\Delta x} (8)$$

quindi:

- $\boldsymbol{rhs}(1) = \theta_1^n + \frac{\Delta t}{\Delta x}\left(K_p^n - K_m^n\right) + \frac{\Delta t}{\Delta x} 2K_m^n \frac{\psi_L^{n+1}}{\Delta x}$;
- $\boldsymbol{a}(1) = 0$;
- $\boldsymbol{b}(1) = \frac{\Delta t}{\Delta x^2}\left(K_p^n + 2K_m^n\right)$;

- $\boldsymbol{c}(1) = -\frac{\Delta t}{\Delta x^2} K_p^n$.

Nel codice Matlab si ha (vengono omessi gli apici relativi allo step temporale):

```
if(i==1)
        Kp = 0.5*(K(i)+K(i+1));
        Km = 0.5*(K(i)+KL);
        %dirichlet (pressure) boundary condition
        a(i) = 0;
        b(i) = dt/dx^2*(2*Km+Kp);
        c(i) = -Kp*dt/dx^2;
        rhs(i) = theta(i) + dt/dx*(Kp-Km) + 2*Km*dt/dx^2*psiL;
```

In modo analogo si opera per la cella al bordo destro:

```
elseif(i==IMAX)
        Kp = 0.5*(K(i)+KR);
        Km = 0.5*(K(i)+K(i-1));
        %dirichlet (pressure boundary condition
        a(i) = -Km*dt/dx^2;
        b(i) = dt/dx^2*(Km+2*Kp);
        c(i) = 0;
        rhs(i) = theta(i) + dt/dx*(Kp-Km) + 2*Kp*dt/dx^2*psiR;
```

e per le altre celle:

```
else
        Kp = 0.5*(K(i)+K(i+1));
        Km = 0.5*(K(i)+K(i-1));
        a(i) = -Km*dt/dx^2;
        b(i) = dt/dx^2*(Km+Kp);
        c(i) = -Kp*dt/dx^2;
        rhs(i) = theta(i) + dt/dx*(Kp-Km);
```

Inizia quindi la parte relativa al metodo Nested Newton.

Calcolo $\psi*$ (nel codice è chiamata $psic$):

$$\psi* = -\frac{1}{\alpha}\left(\frac{n-1}{n}\right)^{1/n} \tag{9}$$

Applicando la decomposizione di Jordan posso scrivere $\boldsymbol{\theta} = \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2$ dove ciascuna componente di $\boldsymbol{\theta}_1$ e $\boldsymbol{\theta}_2$ la posso scrivere come:

$$\begin{cases} \theta_{1,i}(\psi) = \theta_i(\psi) & \theta_{2,i}(\psi) = 0 & \psi \leq \psi* \\ \theta_{1,i}(\psi) = \theta_i(\psi*) + c_i(\psi*)(\psi - \psi*) & \theta_{2,i}(\psi) = \theta_{1,i}(\psi) - \theta_i(\psi) & \psi > \psi* \end{cases} \tag{10}$$

Per la prima iterazione scelgo come valore di primo tentativo $\psi^0 \leq \psi*$ in modo tale che $\boldsymbol{\theta}_2 = 0$ (CasulliZanolli 5.3).

```
psi = min(psi,psic);
```

Allora:

$$\boldsymbol{f}(\boldsymbol{\psi}^{n+1}) = \boldsymbol{\theta}(\boldsymbol{\psi})^{n+1} + \boldsymbol{T}^n \boldsymbol{\psi}^{n+1} - \boldsymbol{rhs}^n = 0 \tag{11}$$

faccio un ciclo su tutte le celle e costruisco $\boldsymbol{f}$: calcolo prima $\boldsymbol{f} = \boldsymbol{\theta} - \boldsymbol{rhs}$ a cui poi vado aggiungere il flusso a seconda della posizione della mia cella poichè la matrice $\boldsymbol{T}$ conosco tramite i vettori $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$ che ho costruito prima. Nel codice:

```
for i=1:IMAX
    %compute the true non linear function f(psi)
    f(i) = Thetaf(psi(i))-rhs(i);
    if(i==1)
        f(i) = f(i) + b(i)*psi(i) + c(i)*psi(i+1);
    elseif(i==IMAX)
        f(i) = f(i)+a(i)*psi(i-1)+b(i)*psi(i);
    else
        f(i) = f(i)+a(i)*psi(i-1)+b(i)*psi(i)+c(i)*psi(i+1);
    end
end
```

A priori al ψ che uso non è la soluzione del mio sistema quindi $\boldsymbol{f} \neq 0$ quindi il residuo di questa iterazione lo calcolo come il modulo di $\boldsymbol{f}$ e lo confronto con la tolleranza fissata.

I valori della suzione vengono memorizzati in in una nuova variabile:

```
psik = psi;
```

Come valore di primo tentativo per la seconda iterazione prendo il massimo tra i valori della suzione che ho ottenuto dall'iterazione precedente e il valore $\psi*$:

```
psi = max(psi,psic);
```

In questo caso ottengo: (CasulliZanolli 5.3)

$$\boldsymbol{\theta}_1 = \boldsymbol{\theta}_1(\boldsymbol{\psi}*) + \boldsymbol{P}(\boldsymbol{\psi}*)(\boldsymbol{\psi} - \boldsymbol{\psi}*) \tag{12}$$

Pertanto il sistema diventa:

$$\boldsymbol{\theta}_1(\boldsymbol{\psi}*)+\boldsymbol{P}(\boldsymbol{\psi}*)\left(\boldsymbol{\psi}^{n+1} - \boldsymbol{\psi}*\right)-\boldsymbol{\theta}_2(\boldsymbol{\psi}^n)+\boldsymbol{Q}(\boldsymbol{\psi}^n)\left(\boldsymbol{\psi}^{n+1} - \boldsymbol{\psi}^n\right)+\boldsymbol{T}^n\boldsymbol{\psi}^{n+1} = \boldsymbol{rhs}^n \tag{13}$$

portando il termine noto a sinistra ottengo:

$$\boldsymbol{fk}(\boldsymbol{\psi}^{n+1}) = \boldsymbol{\theta}_1(\boldsymbol{\psi}*)+\boldsymbol{P}(\boldsymbol{\psi}*)\left(\boldsymbol{\psi}^{n+1} - \boldsymbol{\psi}*\right)-\boldsymbol{\theta}_2(\boldsymbol{\psi}^n)+\boldsymbol{Q}(\boldsymbol{\psi}^n)\left(\boldsymbol{\psi}^{n+1} - \boldsymbol{\psi}^n\right)+\boldsymbol{T}^n\boldsymbol{\psi}^{n+1}-\boldsymbol{rhs}^n = 0 \tag{14}$$

In modo analogo a quanto fatto per $\boldsymbol{f}(\boldsymbol{\psi}^n + 1)$ costruisco $\boldsymbol{fk}(\boldsymbol{\psi}^{n+1})$:

```
for i=1:IMAX
     fk(i)=Theta1(psi(i))-(Theta2(psik(i))+dTheta2(psik(i))*(psi(i)-psik(i)))-rhs(i);
%Tk is frozen
     if(i==1)
          fk(i)=fk(i)+b(i)*psi(i)+c(i)*psi(i+1);
     elseif(i==IMAX)
          fk(i)=fk(i)+a(i)*psi(i-1)+b(i)*psi(i);
     else
          fk(i)=fk(i)+a(i)*psi(i-1)+b(i)*psi(i)+c(i)*psi(i+1);
end
```

la correzione che devo fare al valore di ψ è pari a:

```
    di(i)=dTheta1(psi(i))-dTheta2(psik(i));
```

quindi il sistema che vado a risolvere con l'algoritmo di Thomas è

```
    dpsi = Thomas(a,b+di,c,fk);
    psi = psi(:)-dpsi(:);
```

dove psi è la soluzione aggiornata.