Нина Филипова
Институт по механика, БАН
e-mail: philipova@imbm.bas.bg

РАЗРАБОТВАНЕ НА ИНТЕРАКТИВНА КОМПЮТЪРНА ПРОГРАМА ЗА ПРОЕКТИРАНЕ НА КАПКОВА НАПОИТЕНА СИСТЕМА ЗА МАЛКИ ПЛОЩИ В GUI-MATLAB: I. ИЗИСКВАНИЯ НА СИСТЕМАТА КУЛТУРА-ПОЧВА-ВОДА

DEVELOPMENT OF INTERACTIVE COMPUTER PROGRAM FOR DESIGN OF SMALL SCALE DRIP IRRIGATION SYSSTEM IN GUI – MATLAB: I. CROP WATER REQUIREMENTS

Nina Philipova
Institute of Mechanics, the Bulgarian Academy of Sciences,
Acad. G. Bonchev Str., Bl. 4, 1113 Sofia, Bulgaria,
e-mail: philipova@imbm.bas.bg

Резюме: *Компютърна програма е разработена за проектиране на надземна капкова напоителна система. Тя може да бъде приложена за оразмеряване на системи с малка площ (до 10 ha). Програмата включва две основни части: съобразяване изискванията на системата култура-почва-вода и хидравлични изчисления. Тази публикация разглежда изискванията на системата култура-почва-вода. Програмата е разработена в Graphical User Interface на MATLAB и дава възможности за избор на определени таблични параметри като: агро-физични свойства на почвата, характеристики на съответната култура, климатични данни. Разгледаният пример се отнася за проектиране на капкова система за напояване на овощни дръвчета, по специално ябълки. Програмата позволява потребителя да зададе определени стойности, така например: дебит на капкообразувателя, разстояние между капкообразувателите и разстояние между напоителните крила и т.н. Тя изчислява брутна и нетна поливна норма и всички останали параметри определящи продължителността на поливния процес.*

Ключови думи: *Подпочвено капково напояване**,** изисквания на системата култура-вода, хидравлично оразмеряване, интерактивно програмиране.*

Key words: *drip irrigation**,** crop water requirements, hydraulic design, graphical user interface.*

I. Introduction

The world population will rise from 6.8 billion today to 9.1 billion in 2050, according to the latest UN projections, – "a third more mouths to feed" than there today. Nearly all of the population growth will occur in developing countries [1]. The main challenges which the world agriculture will face in coming decades are usage of natural resources more efficiently and adapting to climate change, producing 70 percent more food for an additional 2.3 billion people by 2050 while at the same time combating poverty and hunger, according to an FAO discussion paper published on 23 September 2009, Rome. Agriculture accounts for 24 % of water abstraction in Europe – up to 80 % in some southern member states – compared to 44 %

abstracted for cooling water in energy production, according to the European Environment Agency. Therefore, the Commission has identified agriculture as the priority sector in which measures to combat water scarcity need to be considered [2]. Fostering water efficient technologies and practices is one of the main policy options of European Commission since 2007 [3].

Drip irrigation offers the most efficient and productive way for applying water and nutrients to crops. The whole systems conveying the water and their emitter devices in particular are very precise. They need a proper design and management. The efforts of some scientists are devoted to improve the components of drip irrigation system: emitters, valves, control equipment. Another group of scientists develops computer programs for performing hydraulic calculations of drip irrigation system. Such products are Hydrocalc (Netaphim), AquaFlow 3.0 Design (Toro), Norveco and etc. All of them offer only hydraulic calculations of a drip irrigation system. A computer program is developed by [4] for hydraulic evaluations of drip irrigation system and its' optimization.

The subject of this paper is to describe the developed program for drip irrigation system design. It is accomplished in Graphical User Interface (GUI) in MATLAB. The program includes two main parts: crop water requirements and hydraulic calculations. It gives convenient way for choosing soil, crop, climate data and emitter characteristics. GUI in MATLAB offers variety of interactive elements: tables, edit texts, radio buttons, pull down menus and push buttons, which can be easily implemented and additionally coded for performing the necessary calculations.

II. Crop Water Requirements I part

1. Total Available Water Capacity, TAWC, (mm).

The methodology, which the program is based on, is with closed connection with adopted knowledge during the attended course on modern irrigation technologies in Israel. Total available water capacity can be calculated according to the following formulae:

$$TAWC = (FC - WP) \times BD \times 10 \quad , \qquad (1)$$

where: *FC* - the Field Capacity, % (weight basis), (Table data); *WP* - Wilting Point, % (weight basis); BD - Bulk Density of the soil, (g/m^3), (Table data according to [5]). We can choose from first window of the program (Fig. 1.) the soil characteristics holding the button CTRL. They are for silty-loam soil the following: $FC = 30$ %, $WP = 12$ % and $BD = 1.38$ g/m^3. *TAWC* can be calculated by the program.

2. Programming Crop Water Requirements I part

```
function uitable1_CellSelectionCallback(hObject, eventdata,
handles)
% hObject    handle to uitable1 (see GCBO)
% eventdata  structure with the following fields (see UITABLE)
%   Indices: row and column indices of the cell(s) currently
selecteds
% handles    structure with handles and user data (see GUIDATA)

handles.selectedCells = eventdata.Indices;
guidata(hObject,handles)
end

% —- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

data = get(handles.uitable1,'data');
```

```
row = handles.selectedCells(:,1);
col = handles.selectedCells(:,2);

for i = 1:length(col)
     if i == 1
          FC = data(row(i),1);
     elseif i == 2
          WP = data(row(i),2);
     elseif i == 3
          BD = data(row(i),3);
     elseif i == 4
          IR = data(row(i),4);
     end
    end

TAWC = (FC-WP)*BD*10;

set(handles.tawc1_Statictext1, 'String', num2str(TAWC)); %Print
TAWC in Statictext1

set(handles.tawc1_Statictext3, 'String', num2str(IR)); %Print IR in
Statictext3

guidata(hObject,handles)

%stores the figure handle of new_cwr2 (second file) here
new_cwr2FigureHandle=new_cwr2

%stores the GUI data from new_cwr2 here
%now we can access any of the data from new_cwr2
handlesData1=guidata(new_cwr2)
end
```

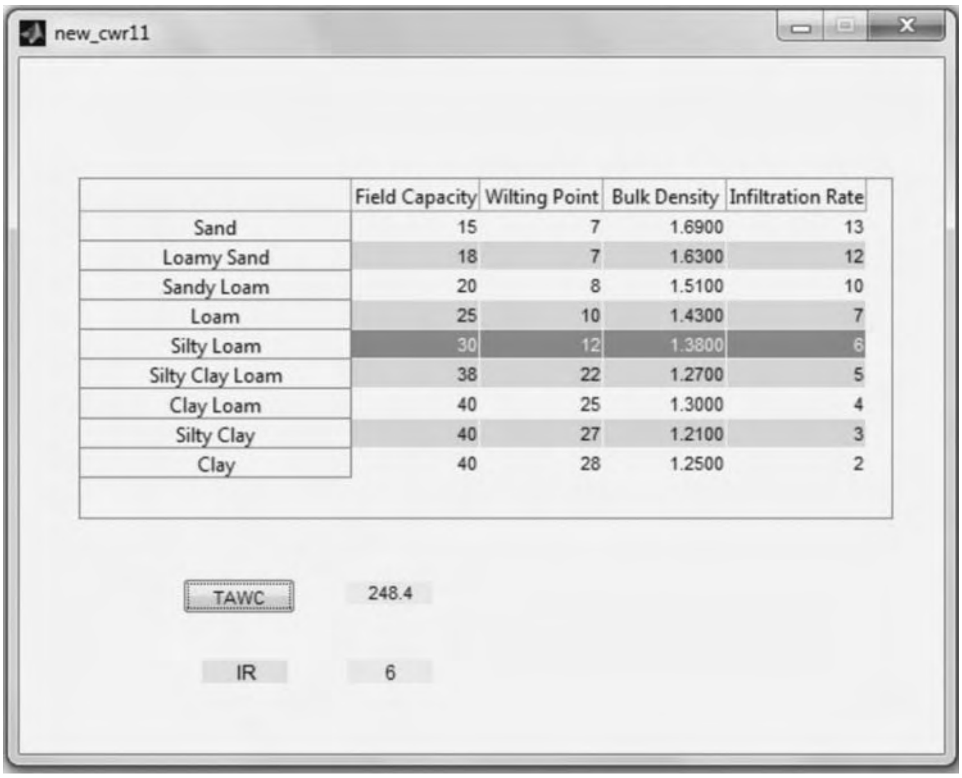

Fig. 1. Crop Water Requirements I part– first window of the program

The user must hold down CTRL and click on every cell of the corresponding row. Pressing the push buttons TAWC and IR will give the following values (Infiltration rate will be used later).

III. Crop Water Requirements II part

1. Readily Available Water (RAW) in the main or effective root zone:

This crop characteristic is determined by means of the next equation:

$$RAW = TAWC \times DRZ \,, \qquad (2)$$

where: *DRZ* - the Design Root Zone or the effective root depth, (m), (Table data according to [6]).

2. Design Net Water Requirement (NWR), (mm):

It can be obtained by calculating the following expression:

$$NWR = RAW \times MAD \times WAR \,, \qquad (3)$$

where: *MAD* - the Maximum Allowable Depletion for the corresponding crop, (%), (Table data according to [6]), *MAD* = 30 - 70 %; *WAR* - the Wetted Area Ratio, %.

3. Wetted Diameter (WD) by emitter:

The equation developed by Zazueta [7] is the following:

$$WD = 0.01\left(\frac{q_e \times DRZ}{IR}\right)^{1/3} , \qquad (4)$$

where: *WD* - the Wetted Diameter by the emitter, (m); q_e - the emitter discharge (l/hr); *IR* - the Infiltration Rate (m/s)

3.1. For non-overlapping wetted area (for wide spaced crops):

The wetted area for non-overlapping drippers can be estimated by the following formulae:

$$WAR = \frac{S_{wet}}{s_e \times s_l} = \frac{\pi \times WD}{4 \times s_e \times s_l} \,, \ \% \qquad (5)$$

where: S_{wet}- the area wetted by emitter, (m^2); s_e- the spacing between the emitters, (m); s_l-the spacing between the laterals, (m).

3.2. For overlapping wetted area (for closed spaced crops):

The wetted area for overlapping drippers can be determined as follows:

$$WAR = \frac{S_{wet}}{s_{e_{reduced}} \times s_l} = \frac{\pi \times WD}{4 \times s_{e_{reduced}} \times s_l} \,, \qquad (6)$$

where: $s_{e_{reduced}}$ - the reduced spacing between the emitters as a result of overlapping.

The area of overlapped circle segment will be kS_{wet}, $k = 0.2$ for 20 % overlapping. We know the wetted diameter and the area to be overlapped. Than, we calculate the apothem and multiplying it by two to obtain the reduced spacing between drippers. For example, for 20% drippers' overlapping $k = 0.2$, $\Theta = 0.5\ \pi$, and $R_{reduced}$ = 0.707R. The reduced emitter spacing will be as follows:

$$s_{e_{reduced}} = 2R_{reduced} \qquad (10)$$

4. Programming Crop Water Requirements II part

```
% —- Outputs from this function are returned to the command line.
function varargout = new_cwr2_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

%stores the figure handle of new_cwr11(first m file) here
new_cwr11FigureHandle=new_cwr11

%stores the GUI data from new_cwr11here
%now we can access any of the data from new_cwr11
handlesData=guidata(new_cwr11FigureHandle)

%store the input text from new_cwr11 into the variable handles.TAWC
handles.TAWC=get(handlesData.tawc1_Statictext1,'String')
%set the static text on new_cwr2 to match the
%input text from handles.TAWC and handles.IR from new_cwr11
set(handles.cwr2_StaticText2,'String',handles.TAWC)
```

```
handles.IR=get(handlesData.tawc1_Statictext3,'String')
set(handles.cwr2_StaticText4,'String',handles.IR)
guidata(hObject, handles);
end

function cwr2_EditText1_Callback(hObject, eventdata, handles)
% hObject    handle to cwr2_EditText1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of cwr2_EditText1 as
text
%        str2double(get(hObject,'String')) returns contents of
cwr2_EditText1 as a double
% q
handles.q=[]
handles.q=str2num(get(hObject,'String')) %Transform string to number
format
guidata(hObject, handles)
end

% —- Executes on button press in cwr2_radiobutton1.
function cwr2_radiobutton1_Callback(hObject, eventdata, handles)
% hObject    handle to cwr2_radiobutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of cwr2_radiobutton1
handles.Se=handles.EmitterSpacing
handles.Sl=handles.LateralSpacing
set(handles.cwr2_StaticText14, 'String', num2str(handles.Se))
guidata(hObject, handles);
end
% —- Executes on button press in radiobutton2.

function radiobutton2_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton2
handles.Se=0.707*handles.EmitterSpacing;
handles.Sl=handles.LateralSpacing;
set(handles.cwr2_StaticText9, 'String', num2str(handles.Se))
set(handles.cwr2_StaticText14, 'String', num2str(handles.Se))
guidata(hObject, handles);
end

% —- Executes on button press in cwr2_pushbutton1.
function cwr2_pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to cwr2_pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% WD
data = get(handles.uitable1,'Data');
Row = handles.selectedCells(:,1);
Col = handles.selectedCells(:,2);

for j = 1:length(Col)
    if j == 1
       DRZ = data(Row(j),1);
        elseif j == 2
        MAD = data(Row(j),2);
    end
end

WD=0.01*((handles.q)*DRZ*3.6*10^6/handles.IR)^(1/3)
handles.WD=WD;
handles.DRZ=DRZ;
handles.MAD=MAD;
set(handles.cwr2_StaticText10, 'String', num2str(WD))
guidata(hObject,handles);
end
```

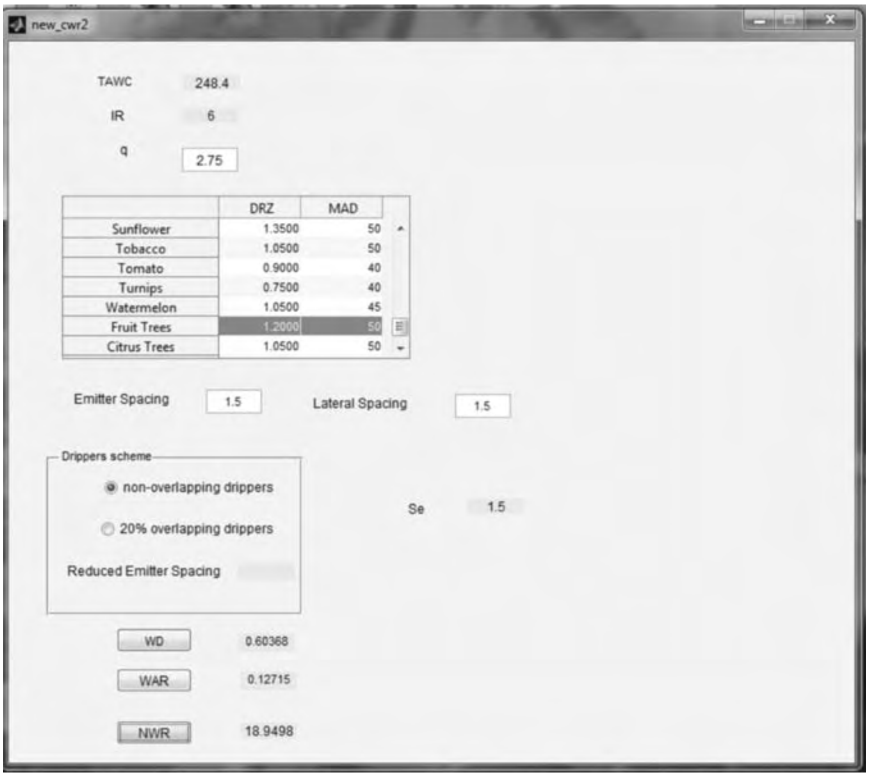

Fig. 3. Crop Water Requirements II part – second window of the program

The user can give a value for emitter discharge and choose and for apples (Fruit trees), and to assume values for the emitter spacing and the lateral spacing. The program either calculates the reduced emitter spacing or equates the value of emitter spacing to the one assumed previously by choosing one of the radio buttons for non-overlapping drippers or for 20 % overlapping drippers. The program calculates the corresponding values of the wetted diameter , the wetted area ratio and the net water requirement, pressing the corresponding push buttons.

IV. Crop Water Requirements III part

1. Gross Water Requirement (GWR), (mm):

It can be calculated using the following expression:

$$GWR = \frac{NWR}{Design\ efficiency, \%}\ , \qquad (11)$$

2. Irrigation Interval (IrI) (Frequency of irrigation):

It is important for irrigation scheduling and is determined by the next ratio:

$$IrI = \frac{NWR}{Etc}\ , \qquad (12)$$

$$Etc = Et_p\, k_c\ , \qquad (13)$$

where: Etc - the peak daily crop evapotranspiration (mm/day); Et_p - the reference crop evapotranspiration (mm/day) (Table data according to [8]); k_c - the crop coefficient. (Table data according to [9]).

3. Application Rate (AR), (mm/hr):

It can be estemated as follows:

$$AR = \frac{q_e}{s_e \times s_l}\ , \qquad (14)$$

4. Duration of Irrigation, (hr).

It can be determined by means of next relationship:

$$Dur.Irr. = \frac{GWR}{AR}\ , \qquad (15)$$

5. Number of shifts

This number depends on ratio between duration of irrigation and the assumed time for system operation:

$$Num.of\ shifts = \frac{Dur.Irr}{Assumed\ \ time\ \ for\ sys.operation}\ , \qquad (16)$$

6. Programming Crop Water Requirements III part

```
function cwr3_EditText1_Callback(hObject, eventdata, handles)
% hObject    handle to cwr3_EditText1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of cwr3_EditText1 as
text
%        str2double(get(hObject,'String')) returns contents of
cwr3_EditText1 as a double
%DesignEfficiency

DesignEfficiency=str2num(get(hObject,'String'))
handles.DesignEfficiency=[]
handles.DesignEfficiency=DesignEfficiency
guidata(hObject,handles)
end

% —- Executes during object creation, after setting all properties.
function cwr3_EditText1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to cwr3_EditText1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
end

% —- Executes on button press in cwr3_pushbutton1.
function cwr3_pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to cwr3_pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%GWR
GWR=(handles.NWR)/handles.DesignEfficiency
handles.GWR=[]
handles.GWR=GWR
set(handles.cwr3_StaticText4, 'String', num2str(handles.GWR))
guidata(hObject, handles)
end
```

We can point out the design efficiency in % in the edit text on the third window of the program and then to obtain values for the gross water requirements, and the application rate by pressing the corresponding push buttons. Then, we have to select one value for the reference evapotranspiration from the table according to the climatic zone and one value for crop coefficient from the next table. The values for crop coefficient are editable and the user can give his own value, which the program will use in the next calculation, but will not save it. Then, we have to double press the push button for . The program will calculate the corresponding values pressing the push buttons of irrigation interval and duration of irrigation. We assume the time for operation and the program calculates the number of plots by pressing the corresponding push button.

V. CONCLUSIONS:

An interactive program is developed for design of small scale drip irrigation systems - up to 10 ha. It is accomplished in GUI - MATLAB and presents the first main part- crop water requirements.

It allows the user in an interactive way to take into account all the necessary soil, crop, climatic data. It will be helpful for designing small scale drip irrigation systems.

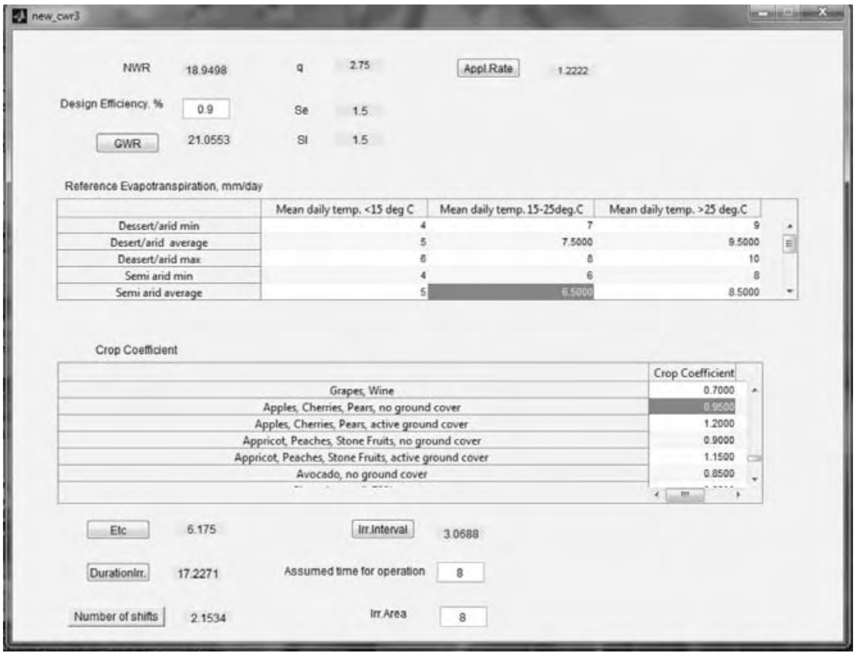

Fig. 4. Crop water requirements III part (third window of the program)

REFERENCES:

1. 2050: A third more mouths to feed, http://www.fao.org/news/story/en/item/35571/icode
2. Future EU water blueprint to focus on savings, http://www.euractiv.com/specialweek-waterpolicy/future-eu-water-blueprint-focus-savings-news-498859.
3. Water Scarcity & Droughts – 2012 EU Policy Review. http://ec.europa.eu/environment/water/ quantity/pdf/non-paper.pdf.
4. Narayanan, R. Steele, D., Scherer, T. Computer Model to optimize Above-ground Drip Irrigation Systems for Small Areas, *Applied Engineering in Agriculture,* **18** (2002), No. 4, 459-469.
5. Ramsey, H., Calculating readily available water. Farm Note 198. Dept. of Agriculture and Food. Government of Western Australia, (2007), p. 1-3.
6. National Engineering Handbook, Irrigation Guide, USDA, http://www. wsi.nrcs.usda.gov/products/W2Q/downloads/Irrigation/National%20Irrigation%20Guide.pdf
7. Sepúlveda, E., F. Zazueta. Wetted bulb dimensions from a trickle source. ABE351. Florida Cooperative Extension Service, Institute of Food and Agricultural Sciences, University of Florida. 2004
8. Irrigation Water Management: Irrigation needs, FAO paper: http://www.fao.org/docrep/S2022E/ S2022E00.htm
9. Crop evapotranspiration - Guidelines for computing crop water requirements, FAO paper, http://www.fao.org/docrep/X0490E/x0490e0b.htm#tabulated kc values