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: II. HYDRAULIC CALCULATIONS

DEVELOPMENT OF INTERACTIVE COMPUTER PROGRAM FOR DESIGN OF SMALL SCALE DRIP IRRIGATION SYSSTEM IN GUI – MATLAB: II. HYDRAULIC CALCULATIONS

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: *Тази публикация разглежда втората част на интерактивна програма за оразмеряване на надземна капкова напоителна система - нейните хидравлични изчисления. В хидравличната част на програмата са заложени осем варианта на разположение на водоизточника и брой напоявани парцели, от които потребителя има възможност да избере един от тях. Тя включва хидравлично оразмеряване на напоителното крило, разпределителния и главния тръбопроводи, както и на помпения възел. Програмата е компилирана за работа в Windows.*

I. Hydraulic calculations I part

1. Hydraulic calculations of lateral

1.1. Head losses due to friction along the lateral:

The Hazen-Williams Equation is used for calculating hydraulic gradient as follows:

$$J_{\%lat} = 1.13\times10^{12}\left(\frac{Q_{lat}}{C_{lat}}\right)^{1.852} D_{lat}^{-4.87} \qquad (17)$$

$$Q_{lat} = N_{e_{latteral}} q_e F_Q \qquad (18)$$

where: $J_{\%lat}$ - the hydraulic gradient of the lateral expressed by promiles %. Q_{lat} - the emitter discharge of the lateral, m^3/s; $N_{elateral}$ - the number of the emitters along a lateral; F_Q=2.7778x10^{-4} - the conversion factor from liters per hour to liters per second; C_{lat} - the Hazen-William coefficient reporting pipe smoothness, C=150 for plastic pipes; D_{lat} - the inside diameter of the drip lateral, m, [10], [11];

The head losses due to friction of the lateral, $\Delta h_{f\ lat}$ (m) are calculated according to the next expression:

$$\Delta h_{f\ lat} = \frac{K_{loc\ lat}\ J_{lat}\ l_{lat}\ F_{c\ lat}}{100} \qquad (19)$$

where: $K_{loc\ lat}$ - coefficient reporting local head losses of the lateral, l_{lat} - the lateral length, m; $F_{c\ lat}$ - the Christiansen coefficient reporting the number of outlets along the drip laterals.

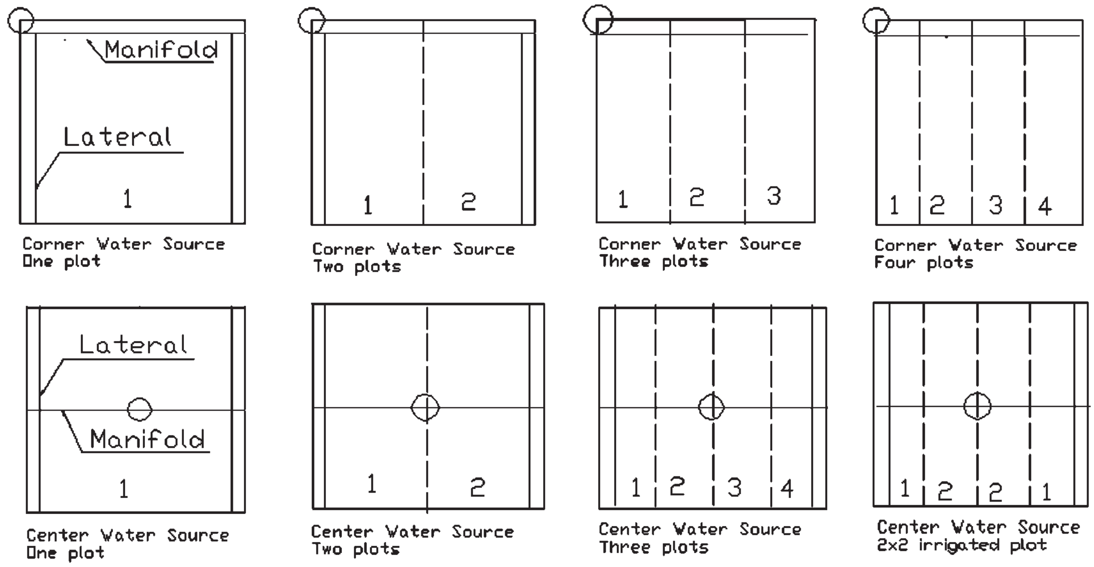

Fig. 5. System layout according to the place of water sorce and the number of plots to be irrigated

It could be calculated according to the formulae:

$$F_{c_{lat}} = \frac{1}{m+1} + \frac{1}{2N_{e_{lateral}}} + \frac{\sqrt{m-1}}{6N^2_{e_{lateral}}} \qquad (20)$$

m=1.852.

1.2. The pressure head by the lateral inlet, $h_{lat\,u}$.

It can be determined as follows:

$$h_{lat\,u} = h_s + 0.75\Delta h_{f\,lat} \pm 0.5\Delta z_{lat} \qquad (21)$$

where: - h_s the nominal pressure head of the dripper, m; Δz - the elevation difference, which is positive for uphill laterals and negative for downhill laterals, $\Delta z_{lat} = s\, l_{lat}$; *s* - the slope of the lateral, %; l_{lat} - the lateral length, m [10], [11];

1.3. The pressure head by the last dripper, $h_{lat\,d}$:

It can be obtained by means of the next expression:

$$h_{lat\,d} = h_{lat\,u} - \Delta h_{f\,lat} \mp \Delta z_{lat} \qquad (22)$$

,

where: Δz - negative for uphill laterals and positive for downhill laterals.

1.4. The pressure head losses along the lateral, h_{lat}:

They are equal to the difference between the mentioned above two lateral hydraulic parameters:

$$\Delta h_{lat} = h_{lat\,u} - h_{lat_d} \qquad (23)$$

where: $h_{lat\,u}$ - the pressure head by the lateral inlet, m; $h_{lat\,d}$ - the pressure head by the last dripper, m;

1.5. Calculation of local head losses of in-line emitters according to the formulae of Provenzano and Pumo

Local head losses due to a coaxial and an uncoaxial emitter can be expressed as a fraction of velocity head immediately downstream the emitter:

$$\Delta h_{loc_e} = \lambda_e \frac{V^2}{2g} = 0{,}065\,\lambda_e \frac{Q}{D^2} \qquad (24)$$

where: $\Delta h_{loc\,e}$ - the local head loss due to a dripper; λ_e - the coefficient synthesizing both local loss due to the contraction and the enlargement of the flow streamlines;

$\frac{V^2}{2g}$ -the velocity head.

Provenzano and Pumo consider λ_e as a function of D/D_e, where *D* - the inside diameter of the dripline; D_e - the "equivalent" emitter diameter, equal to the distance between the inner surface of the emitter and the opposite pipe wall, which can be assumed as an independent value from the Reynolds number.

They have obtained the following empirical relationship:

$$\lambda_e = 0{,}056\left[\left(\frac{D}{D_e}\right)^{17{,}825} - 1\right] \qquad (25)$$

which is true if the inequality

$1{,}00 < \dfrac{D}{D_e} < 1{,}20$ is fulfilled [12].

1.6. Calculation of local head losses according to the formulae of Yella Reddy

Local head losses of on-line dripper are a function of barb protrusion area. Yella Reddy deduced such a formula by applying regression analysis:

$$\Delta h_{loc} = 1{,}525 x_1 + 8{,}427 \qquad (26)$$

where: $\Delta h_{loc\ e}$ - the local head loss due to a dripper in terms of equivalent pipe length [cm]; x_1 is barb protrusion area [mm].

The following formulae is used for obtaining equivalent hydraulic gradient J':

$$J' = J\frac{s + fe}{s} \qquad (27)$$

where: J is the hydaulic gradient from the formula of Darsy - Weisbach; s is the distance between drippers; $fe = h_{loc} x 10^{-2}$, fe - local head loss of a dripper, expressed as equivalent length [13].

2. Hydraulic calculations of the manifold:

2.1. Head losses due to friction along the manifold:

The hydraulic gradient of the manifold is calculated according to the Hazen-Williams equation:

$$J_{\%\,man} = 1.22' \, 10^{10}\left(\frac{Q_{man}}{C_{man}}\right)^{1.852} D_{man}^{-4.87} \qquad (28)$$

$$Q_{man} = N_{lat}\, Q_{lat} \qquad (29)$$

where: Q_{man} - the emitter discharge of the manifold, l/s; N_{lat} - the number of the laterals per a plot; $J_{\%man}$ - the hydraulic gradient of the manifold expressed by promiles %. C_{man} - the Hazen - William coefficient reporting manifold smoothness, C = 150 - for plastic pipes; D_{man} - the inside diameter of the drip manifold, m;

$$\Delta h_{f\,man} = \frac{K_{loc\,man}\, J_{\%\,man}\, l_{man}\, F_{c\,man}}{100} \qquad (30)$$

where: $\Delta h_{f\,man}$ - the head losses due to friction along the manifold, m; $K_{loc\ man}$ - the coefficient reporting local head losses of the manifold, l_{man} - the manifold length, m; $F_{c\ man}$ - the Christiansen coefficient reporting the number of outlets along the manifold.

2.2. The pressure head by the manifold inlet, $h_{man\ u}$:

Analogically to lateral hydraulics, the pressure head by the manifold inlet will be according the next expression:

$$h_{man\,u} = h_{lat_u} + 0.75\Delta h_{f_{man}} \pm 0.5\Delta z_{man} \qquad (31)$$

where: $h_{man\ u}$ - the pressure head by the lateral inlet, m:

2.3. The pressure head by the last lateral, $h_{man\ d}$:

It can also be calculated analogically by means the next formulae:

$$h_{man\,d} = h_{man\,u} - \Delta h_{f\,man} \mp \Delta z_{man} \qquad (32)$$

2.4. The head losses along the manifold, Δh_{man}:

It is equal to the difference between the mentioned above two manifold hydraulic parameters:

$$\Delta h_{man} = h_{man\,u} - h_{man_d} \qquad (33)$$

2.5. Requirement for allowable pressure variations:

The main requirement for allowable pressure variation has to be fulfilled as follows:

$$\Delta h_{lat} + \Delta h_{man} < 0.2\, h_s \qquad (34)$$

3. Hydraulic calculations of the main line.

3.1. The pressure head by the main line inlet, $h_{main\ u}$:

It can be calculated using the previously mentioned formulae referred to the main line:

$$h_{main\,u} = h_{man_u} + \Delta h_{f_{main}} \pm \Delta z_{main} \qquad (35)$$

where: $h_{man\ u}$ - the pressure head by the manifold inlet, m; Δz - the elevation difference, positive for uphill laterals and negative for downhill laterals.

4. Programming Hydraulic calculations I part

```
% —- Executes on selection change in HydroCalc_Lat_popupmenu1.
functionHydroCalc_Lat_ popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to HydroCalc_ Lat_popupmenu1 (see GCBO)
  % eventdata  reserved - to be defined in a future version of MATLAB
  % handles structure with handles and user data (see GUIDATA)
  % Hints: contents = cellstr(get(hObject,'String')) returns HydroCalc_Lat_popupmenu1 contents as cell array
  %contents{get(hObject,'Value')} returns selected item from HydroCalc_Lat_popupmenu1

  % Determine the selected data set.
  str = get(hObject, 'String');
  val = get(hObject,'Value');

  % Set current data to the selected data set.
  switch str{val};
  case 'Center Water Source, Number of plots, p=1' % User selects Center Water Source, Number of plots, p=1


      %Lateral Hydraulics
         L_lat1=(handles.PlotLength-4)/2;
  Ne=(handles.PlotLength-4)/(2*handles.AssumedEmitterSpacing);
         handles.Ne=Ne;


LateralDischarge=(handles.EmitterDischarge)*(handles.Ne)*2.778*10^(-4);                          % Converting l/h to l/s
         handles.LateralDischarge=LateralDischarge;

LateralHydraulicGradient=1.22*10^10*((handles.LateralDischarge/150)^1.852)*(handles.LateralInsideDiameter)^(-4.871);
         handles.LateralHydraulicGradient=LateralHydraulicGradient;

L_lat=(Ne-1)*(handles.AssumedEmitterSpacing)+(handles.AssumedEmitterSpacing)/2;
         handles.LateralLength=L_lat;

         m=1.852;
         F=1/(m+1)+1/(2*handles.Ne)+((m-1)^0.5)/(6*(handles.Ne)^2);

          if handles.row<=12; %Calculate local head losses of in-line emitters according to Eq.24

LateralFrictionHeadLosses=(handles.LateralHydraulicGradient)*(handles.LateralLength)*F/100;
%%%%%
handles.LateralFrictionHeadLosses=LateralFrictionHeadLosses;
  LateralEmitterHeadLosses=0.065*(handles.fe)*(handles.LateralDischarge)/(handles.LateralInsideDiameter)^2;
            handles.LateralEmitterHeadLosses=LateralEmitterHeadLosses;
```

```
LateralSumHeadLosses=1.05*(handles.LateralFrictionHeadLosses+handles.L
ateralEmitterHeadLosses);
              handles.LateralSumHeadLosses=LateralSumHeadLosses;

         else  %Calculate local head losses of on-line emitters accord-
ing to Eq.26

                                          LateralAddHydraulicGradient=
handles.LateralHydraulicGradient*(handles.AssumedEmitterSpacing+handle
s.fe)/handles.AssumedEmitterSpacing;

handles.LateralAddHydraulicGradient=LateralAddHydraulicGradient;

                                            LateralFricEmitHeadLosses=
(handles.LateralAddHydraulicGradient)*(handles.LateralLength)*F/100;
                                    handles.LateralFricEmitHeadLosses=
LateralFricEmitHeadLosses;  % %Friction&Emitter Head Losses
                                          handles.LateralSumHeadLosses
=1.05*(handles.LateralFricEmitHeadLosses);

         end


          deltaZ=(handles.LateralSlope)*(handles.LateralLength);

LateralInletPressureHead=handles.NominalPressureHead+0.75*(handles.Lat
eralSumHeadLosses)- (deltaZ)/2
          handles.LateralInletPressureHead=LateralInletPressureHead;

               LateralEndPressureHead=handles.LateralInletPressureHead-
handles.LateralSumHeadLosses+(deltaZ)/2;
          handles.LateralEndPressureHead=LateralEndPressureHead;

          %Allowable Pressure variation_Lateral Head Losses
                    LateralHeadLosses=handles.LateralInletPressureHead-
handles.LateralEndPressureHead; % Head Losses along the Lateral
          handles.LateralHeadLosses=LateralHeadLosses;

          %Manifold Hydraulics
             L_manifold=(handles.PlotWidth-4)/2;   %???% Length of the
manifold for 1plot
          N_lat=2*2*(handles.PlotWidth-4)/(2*handles.Sl);
          handles.N_lat=round(N_lat)
           handles.ManifoldPlotLength=L_manifold       %Length of the
manifold for 1plot

          ManifoldDischarge=(handles.LateralDischarge)*(handles.N_lat);
%?????
          handles.ManifoldDischarge=ManifoldDischarge;

ManifoldHydraulicGradient=1.22*10^10*((handles.ManifoldDischarge/150)^
  1.852)*(handles.ManifoldInsideDiameter)^(-4.871);
          handles.ManifoldHydraulicGradient=ManifoldHydraulicGradient
                 m=1.852;
                                Fman=1/(m+1)+1/(2*handles.N_lat)+((m-
1)^0.5)/(6*(handles.N_lat)^2)

                                             ManifoldFrictionHeadLosses=
(handles.ManifoldHydraulicGradient)*(handles.ManifoldPlotLength)*Fman/
100;
```

```
        handles.ManifoldFrictionHeadLosses= ManifoldFrictionHeadLosses

                                            ManifoldSumHeadLosses
=1.10*(handles.ManifoldFrictionHeadLosses);
          handles.ManifoldSumHeadLosses=ManifoldSumHeadLosses
         deltaZman=(handles.ManifoldSlope)*(handles.ManifoldPlotLength)

ManifoldInletPressureHead=handles.LateralInletPressureHead+0.75*(handl
es.ManifoldSumHeadLosses)- 0.5*(deltaZman);
          handles.ManifoldInletPressureHead=ManifoldInletPressureHead;

             ManifoldEndPressureHead=handles.ManifoldInletPressureHead-
handles.ManifoldSumHeadLosses+(deltaZman)/2;
          handles.ManifoldEndPressureHead=ManifoldEndPressureHead;

          %Allowable Pressure variation_Lateral & Manifold Head Losses
                  ManifoldHeadLosses=handles.ManifoldInletPressureHead-
handles.ManifoldEndPressureHead;
          handles.ManifoldHeadLosses=ManifoldHeadLosses;


handles.SumLatManifHeadLosses=handles.ManifoldHeadLosses+handles.Later
alHeadLosses;

          AllowablePressureHeadLosses=0.20*handles.NominalPressureHead;

handles.AllowablePressureHeadLosses=AllowablePressureHeadLosses;

          handles.ManifoldLengthArea=(L_manifold)*2;
          handles.LateralLengthPlot=N_lat*(handles.LateralLength)*2;
          handles.LateralLengthArea=N_lat*(handles.LateralLength);
          handles.N_manifolds=2;

          %Main Line Hydraulics
          handles.MainLineLength=0;
          handles.MainLineDischarge=0;
                    SystemDischarge=(handles.ManifoldDischarge*2)*3.6;
%Converting Factor from l/s to cum.m/h_ 3.6
          handles.SystemDischarge=SystemDischarge;

RequiredSystemPressureHead=handles.ManifoldInletPressureHead*2;
         handles.RequiredSystemPressureHead=RequiredSystemPressureHead;

          %Print Calculated Parameters
                    set(handles.HydroCalc_Lat_StaticText17,   'String',
num2str(handles.LateralLength));
                     set(handles.HydroCalc_Lat_StaticText36,'String',
num2str(handles.Ne));
                     set(handles.HydroCalc_Lat_StaticText64,'String',
num2str(handles.LateralDischarge));
                     set(handles.HydroCalc_Lat_StaticText38,'String',
num2str(handles.LateralSumHeadLosses));
                     set(handles.HydroCalc_Lat_StaticText40,'String',
num2str(handles.LateralInletPressureHead));
                     set(handles.HydroCalc_Lat_StaticText42,'String',
num2str(handles.LateralEndPressureHead));
                     set(handles.HydroCalc_Lat_StaticText44,'String',
num2str(handles.LateralHeadLosses));
```

```
                    set(handles.HydroCalc_Lat_StaticText19,   'String',
num2str(handles.N_lat));
                    set(handles.HydroCalc_Lat_StaticText21,   'String',
num2str(handles.LateralLengthPlot));
                    set(handles.HydroCalc_Lat_StaticText23,   'String',
num2str(handles.LateralLengthArea));
                    set(handles.HydroCalc_Lat_StaticText25,   'String',
num2str(handles.ManifoldPlotLength));
                     set(handles.HydroCalc_Lat_StaticText66,'String',
num2str(handles.ManifoldDischarge));
                    set(handles.HydroCalc_Lat_StaticText46,   'String',
num2str(handles.ManifoldSumHeadLosses));
                    set(handles.HydroCalc_Lat_StaticText48,   'String',
num2str(handles.ManifoldInletPressureHead));
                    set(handles.HydroCalc_Lat_StaticText50,   'String',
num2str(handles.ManifoldEndPressureHead));
                    set(handles.HydroCalc_Lat_StaticText52,   'String',
num2str(handles.ManifoldHeadLosses));
                    set(handles.HydroCalc_Lat_StaticText54,   'String',
num2str(handles.SumLatManifHeadLosses));
                    set(handles.HydroCalc_Lat_StaticText56,   'String',
num2str(handles.AllowablePressureHeadLosses));
                    set(handles.HydroCalc_Lat_StaticText27,   'String',
num2str(handles.N_manifolds));
                    set(handles.HydroCalc_Lat_StaticText29,   'String',
num2str(handles.ManifoldLengthArea));
                    set(handles.HydroCalc_Lat_StaticText31,   'String',
num2str(handles.MainLineLength));
                     set(handles.HydroCalc_Lat_StaticText60,'String',
num2str(handles.MainLineDischarge));
                     set(handles.HydroCalc_Lat_StaticText69,'String',
num2str(handles.SystemDischarge))
                     set(handles.HydroCalc_Lat_StaticText71,'String',
num2str(handles.RequiredSystemPressureHead))

          guidata(hObject, handles);

    % handles.current_data = handles.CeWS1;

  case 'Center Water Source, Number of plots, p=2' % User selects Center
Water Source, Number of plots, p=2

           %Lateral Hydraulics
           L_lat1=(handles.PlotLength-4)/2;
           Ne=(handles.PlotLength-4)/(2*handles.AssumedEmitterSpacing);
           handles.Ne=Ne;
          .........
  case 'Center Water Source, Number of plots, p=4'      % User selects
Center Water Source, Number of plots, p=4.

  case 'Center Water Source, Number of plots, p=4, 2x2' % User selects
Center Water Source, Number of plots, p=4,2x2.

  ......

  % Save the handles structure.
  guidata(hObject,handles);

  end
```

Here, we have to assume standard value for emitter spacing on the forth window of the developed program (it is from 0.3 to 1.0 m for NETAFIM). Then, we have to put the plot dimensions and the slope of lateral, manifold and main line in the corresponding edit texts. It is necessary the dripline and emitter data to be selected, the manifold and main line inside diameters to be assumed and the system layout among the 8 cases according to Fig. 5 to be chosen in order the program to calculate the whole lateral, the manifold and the main line hydraulics. It compares the values of the allowable pressure head losses along the lateral and the manifold and these calculated by the program. The calculated sum of the lateral and the manifold pressure head losses is equal to 1.1161 m and the allowable pressure head losses are 2 m. Consequantly, the requirement the former to be less than the latter is satisfied. The program gives the calculated value for the system discharge and the required system pressure head as final results important for the whole system design.

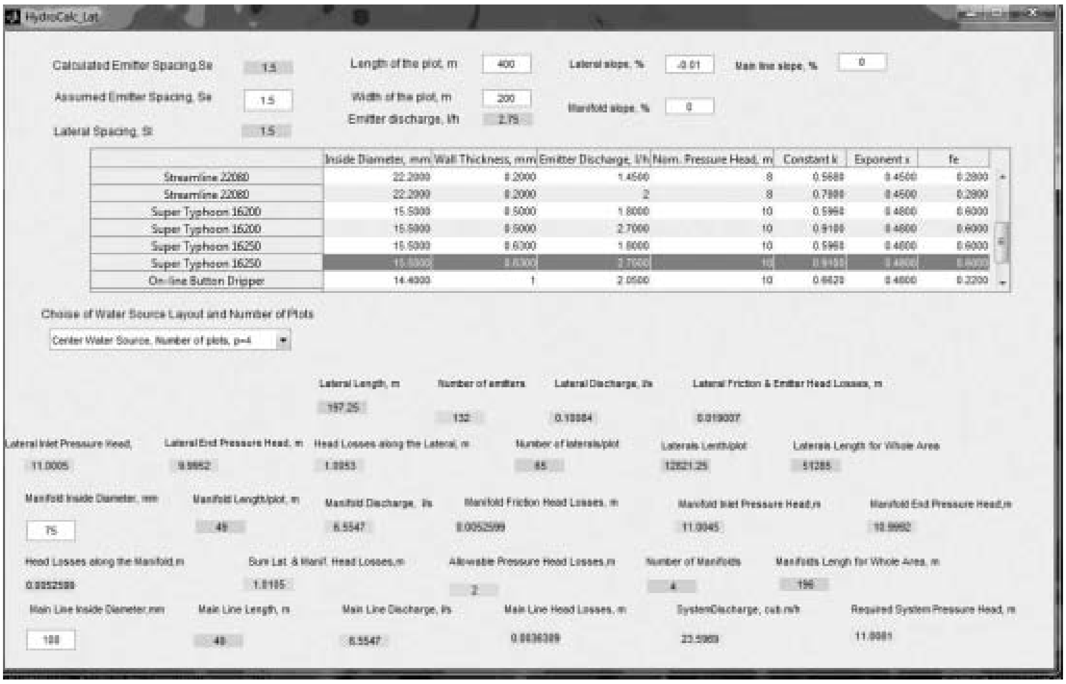

Fig. 5. Hydraulic calculations I part – fourth window of the program

II. Hydraulic calculation II part - Pump characteristics calculation:

1. Total Dynamic Head, TDH, (m):

This characteristic is evaluated as follows:

$$TDH = h_{main_u} + \Delta h_{control\ head} + h_{TSH} \qquad (36)$$

where: h_{mainu}- the pressure head by the main line inlet, m;Δh_{tsh} - the Total Static Head, m; $\Delta h_{control\ head}$ is the pressure head losses in the control head, which are equal to the sum of pressure head losses in the filter, Δh_{filter}, these in the fertilizer tank, $\Delta h_{filter\ tank}$ and these in measuring and control devises $\Delta h_{measur.\&contr.\ device}$:

$$\Delta h_{control\ head} = \Delta h_{filter} + \Delta h_{fertil.\tan k} + \Delta h_{measur.\&contr.devices} \qquad (37)$$

$$\Delta h_{TSH} = \Delta h_{suction\,lift} + \Delta h_{elevetion\,lift} \qquad (38)$$

where: $\Delta h_{suction\ lift}$ - the suction lift of the pump, m; $\Delta h_{elevation\ lift}$ - the elevation lift, m.

2. Pump Power, kW:

This pump characteristic is estimated as follows:

$$P = \frac{Q_p\,TDH}{270\eta} \qquad (39)$$

where: $Q_p = Q_{syst}$ - the discharge of the system, η - the pump efficiency.

3. Programming Hydraulic calculations II part – Pump characteristics calculations

```
function PumpCalc_EditText1_Callback(hObject, eventdata, handles)
% hObject    handle to PumpCalc_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 PumpCalc_EditText1 as text
%        str2double(get(hObject,'String')) returns contents of PumpCalc_EditText1 as a double
SuctionLift=str2num(get(hObject,'String'));
handles.SuctionLift=SuctionLift;
guidata(hObject,handles);
end
% —- Executes during object creation, after setting all properties.
function PumpCalc_EditText1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to PumpCalc_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
function PumpCalc_EditText2_Callback(hObject, eventdata, handles)
% hObject    handle to PumpCalc_EditText2 (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 PumpCalc_EditText2 as text
%        str2double(get(hObject,'String')) returns contents of PumpCalc_EditText2 as a double
ElevationLift=str2num(get(hObject,'String'));
handles.ElevationLift=ElevationLift;
guidata(hObject,handles);
end
% —- Executes during object creation, after setting all properties.
function PumpCalc_EditText2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to PumpCalc_EditText2 (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

function PumpCalc_EditText3_Callback(hObject, eventdata, handles)
% hObject    handle to PumpCalc_EditText3 (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 PumpCalc_EditText3 as text
%        str2double(get(hObject,'String')) returns contents of PumpCalc_EditText3 as a double
ControlHeadHeadLosses=str2num(get(hObject,'String'));
handles.ControlHeadHeadLosses=ControlHeadHeadLosses;
guidata(hObject,handles);
```

```
end

% —- Executes during object creation, after setting all properties.
function PumpCalc_EditText3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to PumpCalc_EditText3 (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
function PumpCalc_EditText4_Callback(hObject, eventdata, handles)
% hObject    handle to PumpCalc_EditText4 (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 PumpCalc_EditText4
as text
%          str2double(get(hObject,'String')) returns contents of
PumpCalc_EditText4 as a double
PumpEfficiency=str2num(get(hObject,'String'));
handles.PumpEfficiency=PumpEfficiency;
guidata(hObject,handles);
end
% —- Executes during object creation, after setting all properties.
function PumpCalc_EditText4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to PumpCalc_EditText4 (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 PumpCalc_pushbutton1.
function PumpCalc_pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to PumpCalc_pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
TotalStaticHead=handles.SuctionLift+handles.ElevationLift;
handles.TotalStaticHead=TotalStaticHead;
set(handles.PumpCalc_StaticText8,                         'String',
num2str(handles.TotalStaticHead));
guidata(hObject,handles);
end
% —- Executes on button press in PumpCalc_pushbutton2.
function PumpCalc_pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to PumpCalc_pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
TotalDynamicHead=(handles.TotalStaticHead)+(handles.RequiredSystemPr
essureHead)+(handles.ControlHeadHeadLosses);
handles.TotalDynamicHead=TotalDynamicHead
set(handles.PumpCalc_StaticText11,                        'String',
num2str(handles.TotalDynamicHead));
guidata(hObject,handles);
end
```

Here, the user can determine the total static head by putting the data for suction and elevation lifts on the fifth window. One can obtain the total dynamic head by pointing out the head losses in control head and then giving the value of pump efficiency so as to have the required pump power, (Kw).

Fig. 6. Pump calculations of drip irrigation system – fifth window of the program

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 this paper presents its second part: hydraulic calculations of surface drip irrigation system. Eight possible cases of system layout are considered and the user can choose one of them from the given popup menu. The program performs all hydraulic calculation of the drip lateral, manifold, main line and pump station. It offers a convenient way for design of small scale drip irrigation systems.

REFERENCES:

1. Azenkot, A. Irrigation System Design, Israel, CINADCO Course Lecture notes, 1998.
2. Keller J., D. Karmeli. Trickle Irrigation Design Parameters, *Transactions of the ASAE,* **17** (1974), No. 4, 678-684.
3. Provenzano G., D. Pumo. Assessing a Local Losses Evaluation Procedure in Drip Integrated Laterals Design. http://afeid.montpellier.cemagref.fr/Mpl2003/AtelierTechno/AtelierTechno/Papier%20Entier/N%C2%B062%20-%20ITALIE_BM.pdf
4. Reddy, Y. Evaluation of On-line Trickle Irrigation Emitter Barb Losses, http://www.ieindia.org/publish/ag/1203/dec03ag3.pdf