Journal of Agriculture and Ecology Research International
3(4): 168-184, 2015; Article no.JAERI.2015.045
ISSN: 2394-1073

SCIENCEDOMAIN *international*
www.sciencedomain.org

SCIENCEDOMAIN

Computer Algorithm for Irrigated Water – Yield Response of Cowpea under Sprinkler Irrigation System

O. T. Faloye[1]* and M. O. Alatise[1]

[1]*Department of Agricultural and Environmental Engineering, Federal University of Technology, P.M.B.704, ZIP 340252, Akure Ondo State, Nigeria.*

Authors' contributions

This work was carried out in collaboration between both authors. Authors OTF and MOA designed the study, wrote the protocol, and wrote the first draft of the manuscript. Author OTF managed the literature searches, analyses of the study performed the structural equation modeling and discuss the conclusion. Both authors read and approved the final manuscript.

Article Information

DOI: 10.9734/JAERI/2015/17298
Editor(s):
(1) Daniele De Wrachien, Department of Agricultural and Environmental Sciences of the State University of Milan, Italy.
Reviewers:
(1) Anonymous, Nigeria.
(2) Ing Kanmogne Abraham, Genie Industriel et Mecanique Ecole Polytechnique Universite de Yaoundé I, Cameroun.
Complete Peer review History: http://www.sciencedomain.org/review-history.php?iid=924&id=37&aid=8875

Original Research Article

Received 7th March 2015
Accepted 31st March 2015
Published 18th April 2015

ABSTRACT

A MATLAB Computer program on cowpea yield was developed to compare the yield obtained from the field under sprinkler irrigation system with the yield predicted by the developed model. Dry season experiment was conducted between January and April of 2014 at Teaching and Research Farm of the Department of Agricultural Engineering, Federal University of Technology, Akure. Soil physical and chemical properties of the experimental site were determined using standard procedures. The cowpea seeds were established on the field and four irrigation water managements were imposed on the crop. An algorithm comprising of existing empirical models from crop production functions were implemented using MATLAB - based computer program. Yield response factor, k_y and elasticity of water production (EWP) were also introduced into the algorithm in order to determine the maximum production of cowpea during the growing season of the experiment. The field seasonal yield, crop evapotranpiration and total water applied were input into the model to validate it so as to obtain corresponding model output (predicted yield). The model

---

*Corresponding author: Email: faloye.temitope@gmail.com;

predicted yield well. The yield results show a good degree of correlation with coefficient of determination r^2 equal to 0.96 and 0.98 for linear and quadratic production functions (ETPF) respectively. The optimized water use and yield of cowpea obtained from the developed model during the growing season were 382.34 mm and 0.996 tons/ha. The total amount of water that resulted to the optimum water use and yield was 446.23 mm. This result implies that 92% of total irrigation water applied during the growing season resulted to the optimum production of the crop. The model, therefore, proved to be useful in estimation of possible irrigation water to maximize yield and crop water use of cowpea.

Keywords: Elasticity of water production; optimum production; modeling; yield response factor; cowpea yield.

NOMENCLATURE

Et_c	*mm of water*	*Crop evapotranspiration*
ET_a	*mm of water*	*Depth of seasonal actual evapotranspiration*
ET_m	*mm of water*	*Maximum evapotranspiration*
ET_{max}	*mm of water*	*Crop maximum evapotranspiration*
Y_a	*tons/ha*	*Actual yield*
Y_{max}	*tons/ha*	*Crop optimal yield*
IRR	*mm of water*	*Seasonal irrigation depth*
IRR_{max}	*mm of water*	*Maximum Seasonal irrigation depth that results in* ET_{max}
TWS_{max}	*mm of water*	*Maximum total water supplied/applied*
$Yield_{max}$	*tons/ha*	*Maximum yield of cowpea*

1. INTRODUCTION

Nigeria has two distinct seasons (rainy season and dry season) with the rainy season, lasting from March to the end of October, and the dry season, lasting from November to March. In the dry season, there is virtually no rain and irrigation remains the only option for crop production. Cowpea is a major crop produced by irrigation, using mostly the sprinkler system. There is competition between municipal, industry users and agriculture for the finite amount of available water. The great challenge for coming decades in the dry season period will be focusing on increase food production by using less water [1]. The lack of water in plant and resulting into water stress has an important effect on water use and yield of crop. The degree of crop responsiveness to water stress can be determined from the crop production functions.

The Crop production functions describe the relationship of crop yield(Y) response to varying levels of water applications. Numerous investigators have also demonstrated the use of crop water production functions in evaluating the economic implications of different levels of crop water use [2] Ayer and Hoyt, [3]; Helweg, [4]; Stegman et al. [5]. Many of the water production functions presented in the literature were developed relating yield (Y) to applied water, which usually includes irrigation water to satisfy crop water requirements in addition to precipitation and stored soil moisture prior to planting. When it became evident that empirical relationship between yield and total amount of water applied on crops cannot be generalized, due to specific geographical locations, soil and water management conditions from which the crops are grown [6]. Indirect pathway was sought relating yield with the field – level water parameter (evapotranspiration). Vaux and Pruitt [7] reported that the yield of a given crop can generally be described as a linear function of cumulative crop evapotranspiration, ET. Although, situations whereby a curvilinear relationship between evapotranspiration and yield exist and have been reported in field studies with different crops by Gulati and Murty [8]. This may be due to increase in evapotranspiration without corresponding increase in yield as a result of excessive water application and can best be represented by quadratic function.

Doorenbos and Kassam [9] introduced the yield response factor to describe the relationship between evapotranspiration (ET) reduction and yield reduction. In the approach of Doorenbos and Kassam [9], yield reductions, and ET deficits are expressed in relative terms based on maximum crop yield (Y_m) and the corresponding

ET at maximum yield (ET_m). Thus, they derived an expression for relative yield decrease as $(1 - Y_a / Y_m) = K_y (1 - ET_a / ET_m)$, where Y_a and ET_a correspond to the actual yield and evapotranspiration (ET), respectively and Y_m and ET_m are maximum yield and maximum crop evapotranspiration, which is attainable for crop grown under optimum condition respectively, and the K_y corresponds to the yield response factor.

On the other hand, the yield response factor (k_y), has been extensively used in research on crop water relations. Therefore, the estimation of the yield response factor of cowpea is important in developing strategies and decisions-making for use by irrigation practitioners for irrigation management under limited water conditions in the study area.

Therefore, there is a need to establish a water use - yield relationship and develop a computer program model using MATLAB based on the empirical linear relationship between the field measurement of crop evapotranspiration and yield to predict yield (output) under varying levels of water (input) applications and also introduce elasticity of water production (EWP) to determine optimum production of cowpea in the study area. The elasticity of water production (EWP) clarifies the relationship between yield and evapotranspiration for optimum production of crops. Crops achieve their maximum production when yield response factor (k_y) is numerically equivalent to elasticity of water production [10].

Considering the importance of water (resources) management, the location of the point where yield response factor (k_y) is numerically equivalent to elasticity of water production of crops should be given utmost attention. Therefore, necessitating the development of a MATLAB – based computer model in an attempt to locate this point of equilibrium for accurate estimation of cowpea optimum production and to determine quantitatively the total seasonal irrigation water depth that resulted to the maximum yield and water use of the crop. The study was done in an attempt to develop a model that predict cowpea yields, and determine the maximum yield and water use by the crop and the seasonal total irrigation water depth that resulted to maximum production of cowpea from the relationship between yield response factor and elasticity of water production.

2. MATERIALS AND METHODS

2.1 Field Experimentation

The experiment was conducted on a sandy clay loam soil at the teaching and research farm of the Federal University of Technology, Akure, Nigeria (7°16'N and longitude 5°13'N). The land was ploughed and harrowed after slashing of shrubs to ensure good soil tilth for crop growth. The layout of the experimental plot was 13 m by 13 m, including the alley way, 1 m wide between treatments. Four irrigation levels were defined with four replicates. Each irrigation level created was 6 m by 6 m with replicates of 2.7 m by 2.7 m making a total of sixteen plots. Cowpea (Vigna unguiculata, L Walp) variety ife brown was planted at the recommended spacing of 30 cm on rows, 60 cm apart. Weeds and insect pests were controlled as necessary using standard procedures.

Irrigation water was applied at each irrigation level for duration of 1 h, 0.8 h, 0.6 h and 0.4 h at treatments (T - 100), (T – 80), (T – 60) and (T – 40) respectively. Two sprinklers each were arranged diagonally at the corner of each irrigation level and making a total of 8sprinkler heads. The sprinklers were allowed to rotate at an angle of 90° and spaced at a distance of 8.49 m at each irrigation level. The sprinklers produced a wetted radius of approximately 6meters to irrigate cowpea in each of the irrigation level at operational pressure of 250 Kpa and average discharge per sprinkler was 0.49 m^3/hr. Control valves were connected to the risers at each irrigation level to stop and regulate the flow of water application at the specified time. Two uniform irrigations were applied to bring the soil to field capacity before planting to encourage seedling establishment. Irrigation depths applied at each irrigation level was predetermined at each irrigation level before sowing cowpea. The irrigation depths were measured using catch cans arranged in each irrigation level. There were twenty (20) cans per irrigation level and the average was estimated over the total area considered (Irrigation level). Rain fall was measured with rain guage and cross-checked with the results obtained using catch cans. Irrigation was scheduled at 4 days between successive irrigation events in all the irrigation levels. The amount of water to meet evapotranspirative demand was applied at treatment (T – 100).

Soil moisture contents were determined in each of the plot bi-weekly at the effective root zone depth of the crop (0 – 0.1, 0.1 - 0.2, 0.2 – 0.3 m) using gravimetric method. The soil moisture content was taken before and after each irrigation. The Soil bulk density (g/cm^3) was determined by the core method [11] using a 10.0cm long by 8.3 cm diameter cylindrical metal core. Runoff and deep percolation were measured using a drainage lysimeter. The drainage lysimeter consisted of drainage and run off system [12].

Evapotranspiration (ET) from each irrigation level (IL) was determined using the soil water balance equation (1).

$$ET_a = P + I + D \pm R \pm \Delta S \quad (1)$$

Where, ET_a is actual evapotranspiration (mm), P is precipitation, I is water applied by irrigation (mm), D is deep percolation below the rooting zone (mm), R is runoff (mm), ΔS is Change in soil water storage (mm). The method involves assessing the incoming and outgoing water flux into the crop root zone over the time interval considered [13].

At maturity, the crops at each irrigation level (IL) were harvested separately and weighed. The grains were threshed from the pods. The yield of cowpea grain was expressed in tons per hectare (ha).

2.2 Description of the Computer Program Prediction Model

The computer program prediction model was developed with the graphical user interface development environment of MATLAB. Algorithm from the existing empirical models (crop production functions) of evapotranspiration – yield functions, and derived linear relationship between seasonal actual evapotranspiration and yield were used to implement (code) the MATLAB based computer program. The algorithms used from the existing empirical models (crop production functions) and derived model include the following:

2.2.1 The linear evapotranspiration production function (ETPF)

The linear evapotranspiration production function (ETPF) model from which the yield is calculated takes the form;

$$Y = a_1 + b_1 ET \quad (2)$$

Where

Y = Measured yield from the field (tons/ha)
ET = Seasonal actual evapotranspiration (mm)
a_1 and b_1 are constants

2.2.2 The quadratic evapotranspiration production function (ETPF)

The yield from the quadratic evapotranspiration production function (ETPF) of water production function is calculated as;

$$Y = a_2 + b_2 ET + c_2 ET^2 \quad (3)$$

Where c_2 is a constant

2.2.3 The slope of the relative yield (Y) reduction versus relative evapotranspiration (ET) deficit

The yield response factor k_y is the slope of the relative yield reduction versus relative evapotranspiration deficit as described by Doorenbos and Kassam [9]. The relationship is expressed mathematically below and yield response factor is calculated as;

$$\left(1 - \frac{Y_{act}}{Y_{\max}}\right) = K_y \left(1 - \frac{ET_{act}}{ET_{\max}}\right) \quad (4)$$

Where

Y_{max} = Maximum Yield
Y_{act} = Actual harvested yield
ET_{act} = Actual evapotranspiration
ET_{max} = Maximum evapotranspiration
K_y = Yield response factor
$\left(1-\frac{Y_a}{Y_m}\right)$ = Seasonal yield relative reduction
$\left(1-\frac{ET_a}{ET_m}\right)$ = Seasonal actual relative evapotranspiration (mm)

2.2.4 The elasticity of water production

The elasticity of water production is the responsiveness of yield to varying water applications. It provides means of comparing relative change in yield with relative change in evapotranspiration [6] and of determining the optimum production of crop at the point where yield response factor is numerically equivalent to the elasticity of water production (EWP). The MATLAB was used and program to locate this point of equilibrium and output the corresponding values. It can be expressed mathematically as;

$$EWP = \frac{dY/Y}{dET/ET} \tag{5}$$

Where

EWP = Elasticity of Water Production (EWP)
$dY =$ Change in yield
$Y =$ Yield (tons/ha)
$ET =$ Seasonal evapotranspiratrion (mm)
$dET =$ Change in evapotranspiration

The elasticity of water production was applied to derive the linear relationship between yield and seasonal actual evapotranspiration obtained during the growing season. The elasticity of water production (EWP) is calculated as described by Liu et al. [10].

$$\text{EWP} = \frac{b_1 ET}{a_1 + b_1 ET} \tag{6}$$

Where

EWP = Elasticity of water production
ET = Seasonal evapotranspiration (mm)
a_1 and b_1 are constants

2.3 Model Implementation

The algorithm for solving the model was implemented in MATLAB (computer software) as operational tool. The solution to the model was built with the graphical user interface development environment (GUIDE) of the MATLAB and the series of (MATLAB computer program) codes that enables the model to run was written. The design of the model in the graphical user interface development (GUIDE) took the form and outlook described and illustrated in Fig. 1.

2.4 Model Input and Output

Input data required by the model include: Four data - set of cowpea seasonal yield obtained from the field at each irrigation level, total water applied and the seasonal actual evapotranspiration measured from the field at each irrigation level using soil water balance method. The model gives output of elasticity of water production (EWP), yield response factor (k_y), yield for the linear and quadratic evapotranspiration production functions (ETPF), optimal yield, optimal seasonal actual evapotranspiration and optimal irrigation water that would maximize cowpea yield during the growing season of the crop.

2.5 Data Analysis

The measured and predicted yield evaluated the model using regressions analysis, while the coefficient of determination (r^2), root mean square error (RSME) and the mean absolute error (MAE) were obtained. These statistical parameters were performed using SPSS and Micro soft Excel to quantify the degree of under/over prediction by the developed model.

3. RESULTS AND DISCUSSION

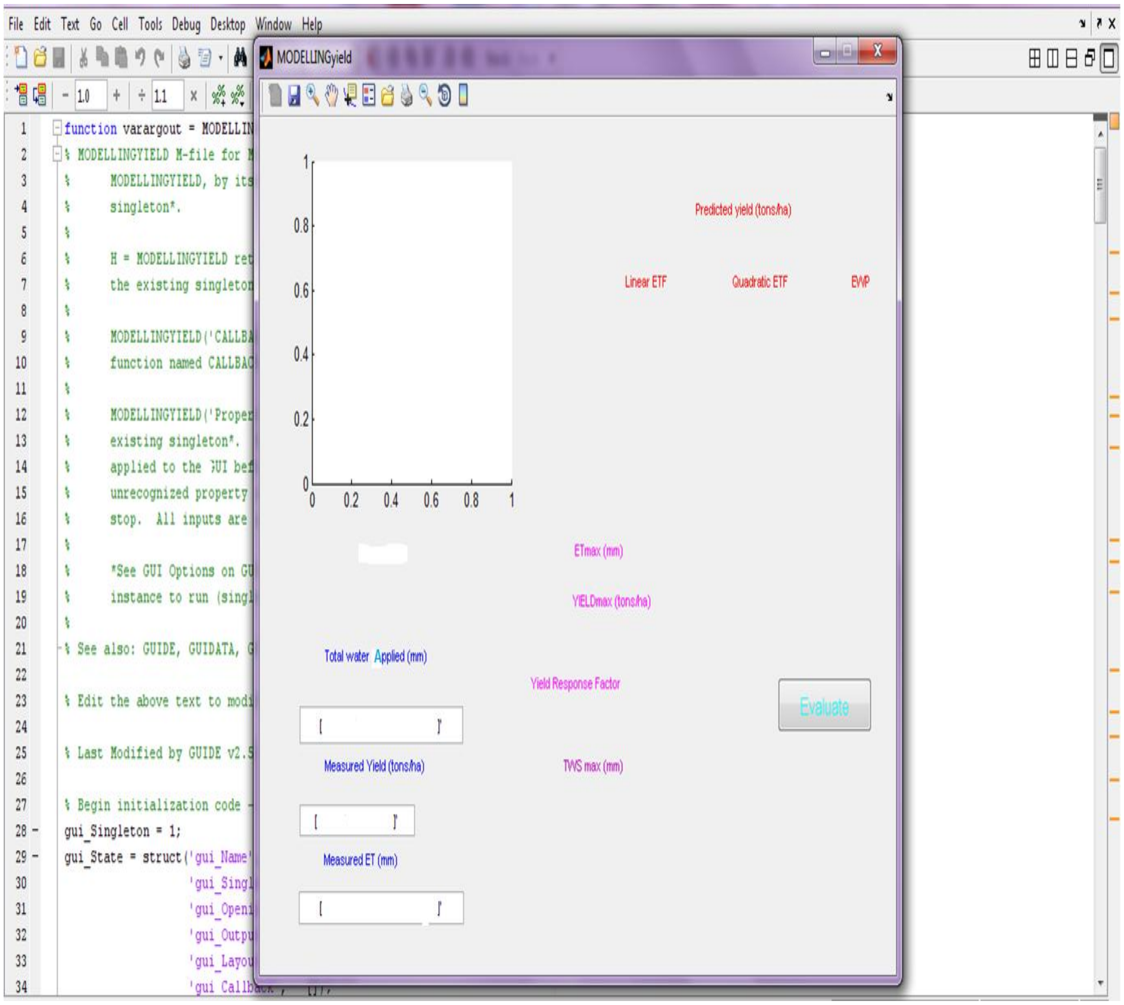
3.1 Field Seasonal Total Water Applied, Actual Evapotranspiration and Yield in the Growing Period of 2014

The total number of irrigation, the amount of irrigation and seasonal actual evapotranspiration values obtained for cowpea using water balance equation during the experiment were presented in Table 1.

Means of yield in each column bearing the same letter are not significantly different at the 5% level of probability by Tukey's test.

In treatment (T–100), the amount of total irrigation water applied and actual seasonal evapotranspiration (ET_a) values were 463.16 mm and 397.52 mm. As expected, the highest seasonal actual evapotranspiration ET_a was obtained in the (T–100) obviously owing to an adequate soil water supply during the growing season. Other treatments underwent water deficits produced lower actual evapotranspiration (ET_a). The result from the research study shows that treatment (T-40) that received the lowest water application of 345.71 mm had the lowest seasonal actual evapotranspiration of 295.96 mm. The seasonal actual evapotranspiration ET_a of the full irrigated (T–100) cowpea in this study was low compared to a value 457.70mm reported by Hashim et al. [14] and much higher than a value of 262.50 mm reported by Adekalu and Okunade [15]. This may be as a result of high and frequent rainfalls that accompany the irrigation events during the late season of the growing season when the crop does not need much water and this period serves as the onset of raining season. A total of 5 irrigation events were recorded during this period resulting to 76.51 mm of rainfall. Thus, it leads to an increase in the crop (cowpea) seasonal actual evapotranspiration during the growing period.

Grain yields measured from the field was observed to decrease with decrease in water application. The irrigation level that received the highest amount of water had the highest yield of 1.06 tons/ha and the treatment that received lowest water application had the lowest yield of 0.71 tons/ha. Statistical evaluation of the experimental data obtained for grain yield in all



```
function varargout = MODELLIN
% MODELLINGYIELD M-file for M
%      MODELLINGYIELD, by its
%      singleton*.
%
%      H = MODELLINGYIELD ret
%      the existing singleton
%
%      MODELLINGYIELD('CALLBA
%      function named CALLBAC
%
%      MODELLINGYIELD('Proper
%      existing singleton*.
%      applied to the GUI bef
%      unrecognized property
%      stop.  All inputs are
%
%      *See GUI Options on GU
%      instance to run (singl
%
% See also: GUIDE, GUIDATA, G

% Edit the above text to modi

% Last Modified by GUIDE v2.5

% Begin initialization code -
gui_Singleton = 1;
gui_State = struct('gui_Name'
                   'gui_Singl
                   'gui_Openi
                   'gui_Outpu
                   'gui_Layou
                   'gui_Callb
```

Fig. 1. Snap shot for the model program, input and output

Table 1. Total number of irrigation, precipitation, total water applied and seasonal ET values of cowpea in the growing period of 2014

Treatments	Number of irrigation	Irrigation (mm)	Total water applied (mm)	Seasonal ET(mm)	Yield (tons/ha)
T-100	11	223.11	462.88	397.28	1.06a
T-80	11	193.85	433.90	371.48	0.95a
T-60	11	141.39	381.44	335.38	0.89a
T-40	11	105.66	345.71	295.96	0.71b

the treatment plots during the growing season showed that grain yield was significantly affected by water applications at 5% level of significance using tukey's test.

3.2 Model Validation

After the experimental data of cowpea grain yield, seasonal actual evapotranspiration and total amount of water applied in each irrigation level have been obtained from the field, the MATLAB was coded and the model was developed with the graphical user interface development environment (GUIDE) of the MATLAB. The model was therefore validated with four data-set of total seasonal water applied, actual evapotranspiration and yield measured from the field at each treatment. Table 2 shows the yield output from the model and the measured yield obtained at the four irrigation water management imposed on the crop using the linear and quadratic evapotranspiration functions.

Statistical parameters were used to assess the model accurately. The parameters are coefficient of determination (r^2), root mean square error (RMSE) and mean absolute error (MAE). The model predicted yield accurately. The coefficients of determination for both linear and quadratic ETPF are 0.96 and 0.98 respectively which shows that there is strong degree of correlation between the predicted yields and measured yields. The quadratic ETPF gave a better yield prediction, but the linear ETPF model is considered more practical and simple is in application. The goodness-of-fit statistics MAE and RMSE used for the comparison of model estimates and observed yield values are presented in Table 3.

The higher the coefficient of determination (r^2) value, the lower the MAE and RMSE values, and the more accurate are the output from the model. The MAE and RMSE values for both quadratic ETPF and linear ETPF indicate that the quadratic ETPF model has the best fit and more accurate in the yield prediction. But, the linear ETPF model is considered more practical and simple in application.

3.3 Relationship between Yield Response Factor and Elasticity of Water Production

Yield response factor (k_y) was determined from the crop production function and implemented in the algorithm of the developed model using MATLAB. The slope of the relative yield reduction and relative evapotranspiration reduction is the yield response factor (Fig. 4). The yield response factor, k_y of cowpea output from the model was 1.24. The value of yield response factor, k_y obtained shows that cowpea is sensitive to water stress. The value compares favourably with value of 1.15 reported by Doorenbos and Kassam [9] for bean and peas FAO [16].

Also, the Elasticity of Water Production (EWP) clarifies the relationship between yield and evapotranspiration. Crops achieve maximum yield (Ym) or optimal yield when the yield response factor, ky is numerically equivalent to elasticity of water production EWP [10]. The MATLAB – based computer program was used to locate the point where elasticity of water production (EWP) is equal to yield response factor, k_y. At this point of equilibrium, the optimal amount of water applied and water use (evapotranspiration).

Fig. 2 and Fig. 3 show the relationship between the measured yields and the predicted yields for both linear and quadratic ETPF respectively that would produce an optimal yield was output from the developed model. Table 4 shows the model output of elasticity of water production from the MATLAB obtained for cowpea.

Table 2. Model validation for the linear and quadratic evapotranspiration production functions

Treatments	Measured yield (tons/ha)	Linear (ETPF) predicted yield (tons/ha)	Qudratic (ETPF) predicted yield (tons/ha)
T - 100	1.06	1.06	1.05
T – 80	0.95	0.97	0.98
T – 60	0.89	0.85	0.87
T – 40	0.71	0.73	0.72

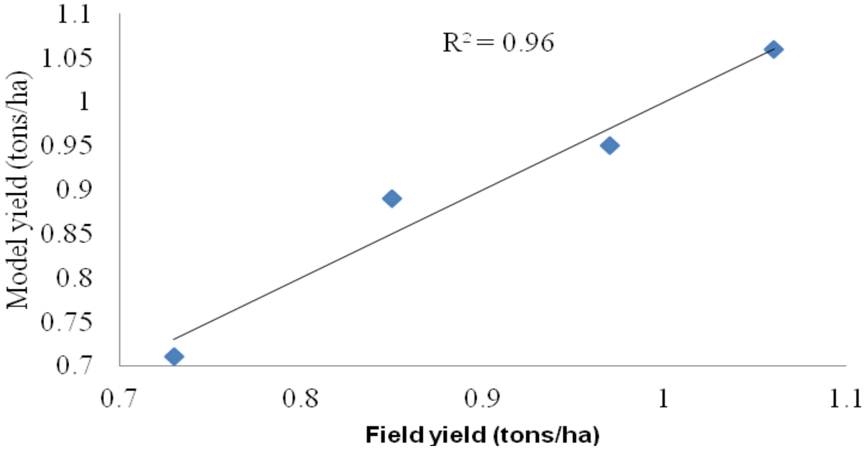


Fig. 2. Cowpea field yield versus model yield for linear evapotranspiration function (ETF)

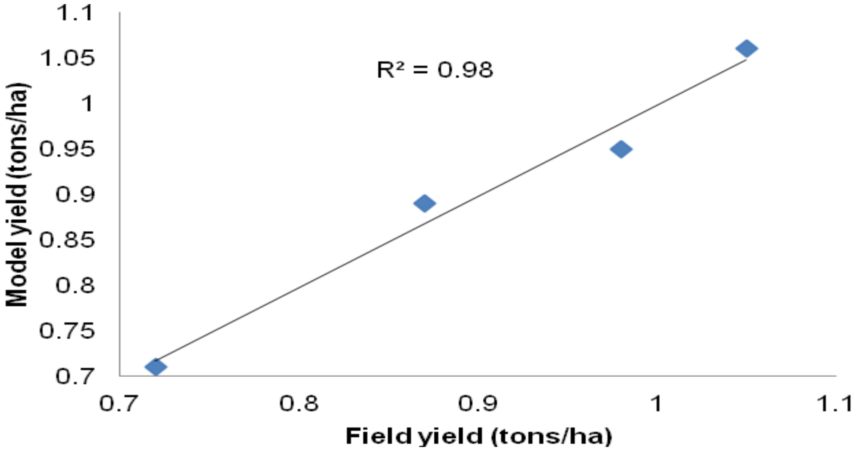


Fig. 3. Cowpea field yield versus model yield for quadratic evapotranspiration function (ETF)

Table 3. Statistical parameters to compare the predicted and field measured yield values

Model	MAE	RMSE	R2
Quadratic ETPF	0.0175	0.0194	0.98
Linear ETPF	0.020	0.0245	0.96

The results presented in Table 4. shows that yield response factor value obtained falls in between the values of elasticity of production (EWP) of treatments (T–100) and (T–80). Linear interpolation between the values of elasticity of water production (EWP) obtained for treatments (T – 100) and (T – 80) was done programmatically using MATLAB to locate the point where elasticity of water production (EWP) and yield response factor (K_y) are numerically equal. Therefore, at this interpolated point, the cowpea optimum water use (evapotranspiration), and yield were 382.337 mm and 0.996 tons/ha respectively, during the growing season of the crop at the experimental site. The seasonal total water applied (including precipitation) and seasonal total irrigation depth that resulted to the optimum production were 446.23 mm and 206.07 mm respectively. This result implies that a total of 16.93 mm irrigation water applied would be saved, which is equivalent to 8% of total irrigation water applied during the growing season and that a total of 92% of total irrigation water applied resulted to the optimum production of the crop using the relationship between elasticity of water production and yield response factor.

Fig. 5 illustrates the relationship between relative yield (ET_r), elasticity of water production (EWP) and relative seasonal actual evapotranspiration of cowpea.

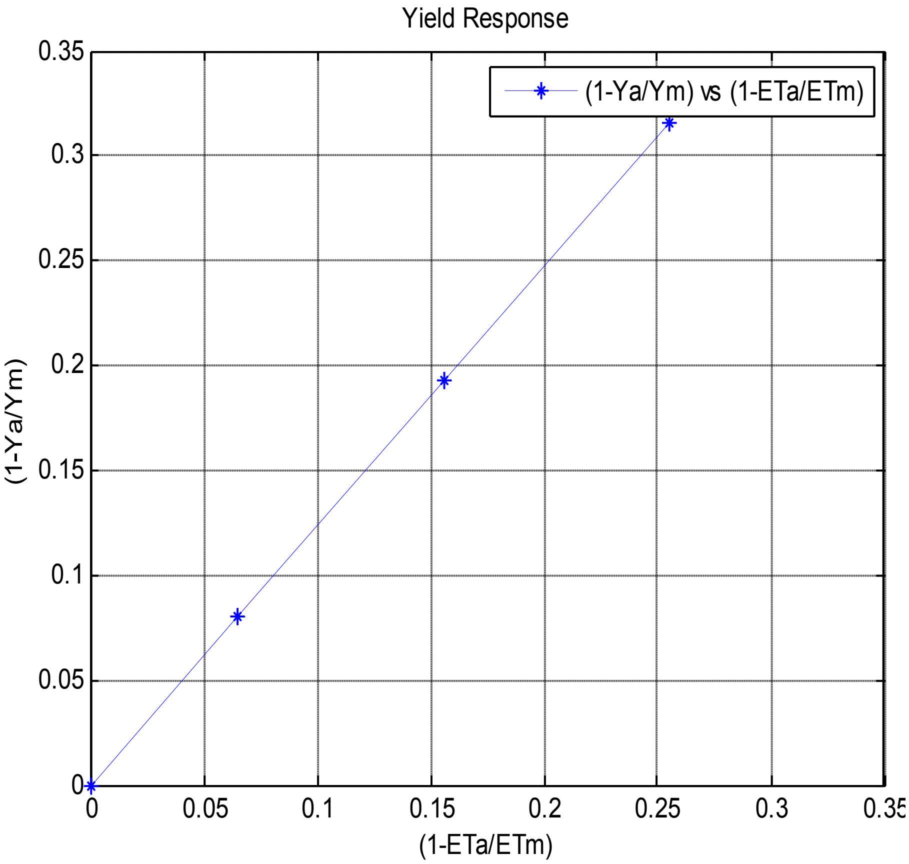
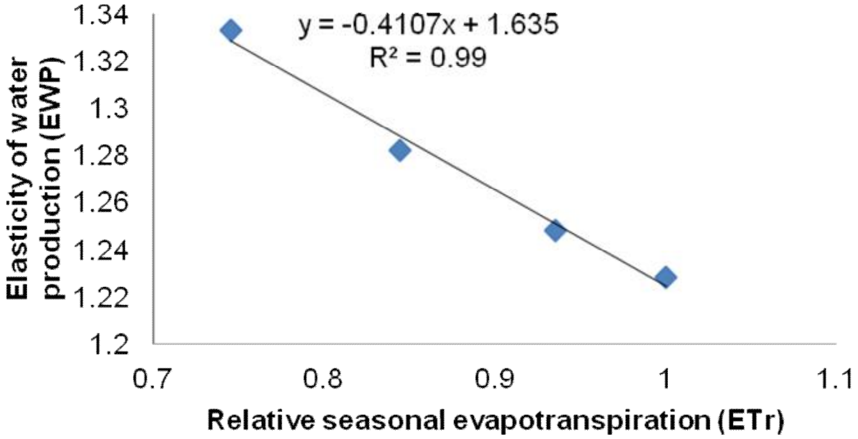


Fig. 4. MATLAB fitted line Output for the relationship between relative yield reduction and relative evapotranspiration reduction for yield response factor estimation

Table 4. Elasticity of water production, seasonal actual evapotranspiration and cowpea seasonal grain yield in the study area during the growing season

Treatments	EWP	Seasonal actual evapotranspiration (mm)	Cowpea seasonal grain yield (tons/ha)
T -100	1.2286	397.28	1.06
T - 80	1.2483	371.48	0.95
T - 60	1.2826	335.38	0.89
T - 40	1.3328	295.96	0.71

The coefficient of determination (r^2) between elasticity of water production (EWP) and relative seasonal actual evapotranspiration (ET_r) is 0.99, therefore indicating that there is a strong relationship between elasticity of water production (EWP) and cowpea relative seasonal actual evapotranspiration. Similar result was obtained for the relationship between relative seasonal yield and relative cowpea seasonal evapotranspiration with coefficient of determination r^2 equal to 0.97. Fig. 5 shows that elasticity of water production obtained for cowpea at the end of the growing season increases as the relative evapotranspiration reduces and the relative yield increases as relative evapotranspiration increases. The intercept on the axis is positive for the linear ETPF model of cowpea for the EWP versus ET_r and negative for the relative yield, Y_r versus and relative evapotranspiration, ET_r. These trends are in agreement with the submission given by [9,10] for the similar established relationships.



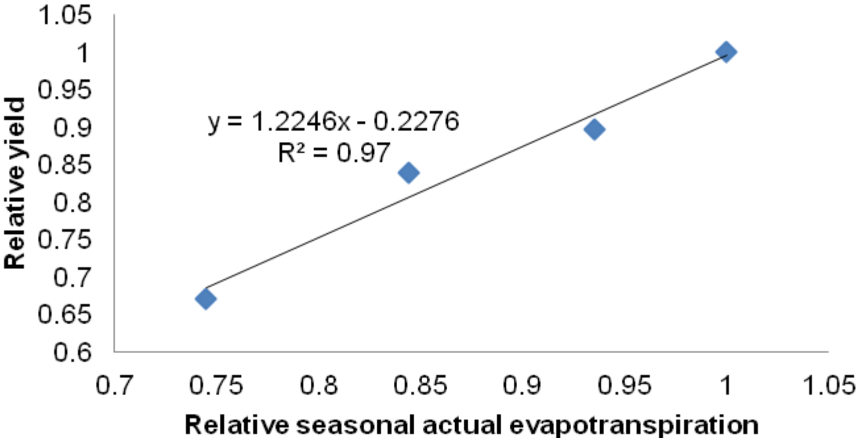


Fig. 5. Relationship between relative yield (ETr), elasticity of water production (EWP) and relative seasonal actual evapotranspiration of cowpea

4. CONCLUSION

A yield prediction model comprising of linear and quadratic evapotranspiration functions were implemented using MATLAB (Computer programming language). The model was validated with four data – set of yield, total amount of water applied and seasonal actual evapotranspiration measured from the four irrigation levels on the field. There was good agreement between measured and predicted yield with reasonable measure of accuracy under the four irrigation water managements imposed on the crop (cowpea) during the growing season.

The optimum production of cowpea was determined in the study area during the growing season at the point where elasticity of water production (EWP) of cowpea is numerically equivalent to the yield response factor. The MATLAB (computer program) accurately locate this point where elasticity of water production (EWP) of cowpea is numerically equivalent to the yield response factor and output the corresponding optimum yield and the cowpea evapotranspiration. The amount of water that resulted to this maximum production was accurately located and determined using MATLAB. The developed model is therefore useful in estimating possible seasonal depth of irrigation water that would maximize water use (evapotranspiration) and yield of cowpea under varying water applications using the relationship between yield response factor and elasticity of water production.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. FAO. Deficit irrigation practices. FAO Water Report No. 22, Rome; 2002b.
2. English MJ. Deficit irrigation. I. Analytical framework. J. Irrig. Drain Eng. 1990; 116(3):399-412.
3. Ayer HW, Hoyt PG. Crop-water production functions: Economic implications for Arizona. Technical Bulletin No. 242. Agriculture and Experiment Station, College of Agriculture, The University of Arizona, Tucson, AZ. 1981;22.
4. Helweg OJ. Functions of crop yield from applied water. Agron. J. 1991;83(4):769-773.
5. Stegman EC, Musick JT, Stewart JI. Irrigation water management. In: Jensen, M.E. (Ed.), Design and operation of Farm Irrigation Systems, Monograph No.3. ASAE, St. Joseph, MI. 1980;763-816.
6. Hexem RW, Heady EO. Water production functions for irrigated agriculture. Iowa State University Press, Ames, IA. 1978; 215.
7. Vaux Jr HJ, Pruit WO. Crop-water production functions. In: Hillel, D.(Ed.), Advance In Irrigation. Academic Press, New York. 1983;2:61-97.
8. Gulati HS, Murty VVN. A model for optimal allocations of canal water based on crop production functions. Agric. Water Manage. 1979;2(1):79-91.
9. Doorenbos J, Kassam AH. Yield response to water. FAO Irrigation and Drainage Paper 33, Rome, Italy. 1979;193.
10. Liu WZ, Hunsaker DJ, Xie XQ, Wall GW, Li YS. Interrelations of yield, evapotranspiration and water use efficiency from marginal analysis of water production functions. Agricultural Water Management. 2002;56(2002):143-151.
11. Blake GR, Hartage KH. Bulk density. In: A. Klute (ed). Methods of Soil Analysis Part 1. 2nd ed. Agron. Monog. 9, ASA, Madison WI. 1986;363 - 375.
12. Igbadun HE. Estimation of crop water use of rain-fed maize and groundnut using mini-lysimeters. Pacific Journal of Science and Technology. 2012;13(1):527-535.
13. Hillel D. Environmental soil physics. Academy Press, New York. 1998;19-123.
14. Hashim MAA, Siam N, Al-Dosari A, Asl-Gaadi KA, Patil VC, Rangaswamy M, Samdani MS. Determination of water requirement and crop water productivity of crops grown in the makkah region of Saudi Arabia. Australian Journal of Basic and Applied Sciences. 2012;6(9):196-206.
15. Adekalu KO, Okunade DA. Effect of irrigation amount and tillage system on yield and water use efficiency of cowpea. Communication in Soil Sci. and Plant Analysis. 2006;37:225-228.
16. FAO. Crop Water Management-Alfalfa. Land and Water Development Division, FAO, Rome; 2002a.

APPENDIX

Cowpea Yield Modeling Program

```
function varargout = MODELLINGyield(varargin)
% MODELLINGYIELD M-file for MODELLINGyield.fig
% MODELLINGYIELD, by itself, creates a new MODELLINGYIELD or raises the existing
% singleton*.
%
% H = MODELLINGYIELD returns the handle to a new MODELLINGYIELD or the handle to
% the existing singleton*.
%
% MODELLINGYIELD('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in MODELLINGYIELD.M with the given input arguments.
%
% MODELLINGYIELD('Property','Value',...) creates a new MODELLINGYIELD or raises the
% existing singleton*.  Starting from the left, property value pairs are
% applied to the GUI before MODELLINGyield_OpeningFcn gets called.  An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to MODELLINGyield_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%  instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help MODELLINGyield
% Last Modified by GUIDE v2.5 03-Jun-2014 00:20:59
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @MODELLINGyield_OpeningFcn, ...
                   'gui_OutputFcn',  @MODELLINGyield_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before MODELLINGyield is made visible.
function MODELLINGyield_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to MODELLINGyield (see VARARGIN)
```

```
% Choose default command line output for MODELLINGyield
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
% UIWAIT makes MODELLINGyield wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command line.
function varargout = MODELLINGyield_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;
function iri_input_Callback(hObject, eventdata, handles)
% hObject handle to iri_input (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 iri_input as text
%        str2double(get(hObject,'String')) returns contents of iri_input as a double
set(handles.plot_button,'Enable','off')
try
   Irrigation = eval(get(handles.iri_input,'String'));
   if ~isnumeric(Irrigation)
      % Irrigation is not a number
      set(handles.plot_button,'String','Irrigation is not numeric')
   elseif length(Irrigation) < 2
      % Irrigation is not a vector
      set(handles.plot_button,'String','Irrigation must be vector')
   elseif length(Irrigation) > 1000
      % Reading is too long a vector to plot clearly
      set(handles.plot_button,'String','t is too long')
   elseif min(diff(Irrigation)) < 0
      % Irrigation is not monotonically increasing
      set(handles.plot_button,'String','Stress must increase')
   else
      % All OK; Enable the Plot button with its original name
      set(handles.plot_button,'String','Plot')
      set(handles.plot_button,'Enable','on')
      return
  end
  % Found an input error other than a bad expression
  % Give the edit text box focus so user can correct the error
  uicontrol(hObject)
 catch EM
   % Cannot evaluate expression user typed
   set(handles.plot_button,'String','Cannot plot Irrigation')
   % Give the edit text box focus so user can correct the error
   uicontrol(hObject)
end
```

```
% --- Executes during object creation, after setting all properties.
function iri_input_CreateFcn(hObject, eventdata, handles)
% hObject handle to iri_input (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
function stress_input_Callback(hObject, eventdata, handles)
% hObject handle to stress_input (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 stress_input as text
%        str2double(get(hObject,'String')) returns contents of stress_input as a double
% --- Executes during object creation, after setting all properties.
% hObject handle to stress_input (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 stress_input as text
%        str2double(get(hObject,'String')) returns contents of stress_input as a double
set(handles.plot_button,'Enable','off')
try
   Stress = eval(get(handles.stress_input,'String'));
   if ~isnumeric(stress)
      % stress is not a number
      set(handles.plot_button,'String','Stress is not numeric')
   elseif length(Stress) < 2
      % Reading is not a vector
      set(handles.plot_button,'String','Stress must be vector')
   elseif length(stress) > 1000
      % Reading is too long a vector to plot clearly
      set(handles.plot_button,'String','t is too long')
   elseif min(diff(Stress)) < 0
      % Reading is not monotonically increasing
      set(handles.plot_button,'String','Stress must increase')
   else
      % All OK; Enable the Plot button with its original name
      set(handles.plot_button,'String','Plot')
      set(handles.plot_button,'Enable','on')
      return
  end
  % Found an input error other than a bad expression
  % Give the edit text box focus so user can correct the error
  uicontrol(hObject)
 catch EM
   % Cannot evaluate expression user typed
   set(handles.plot_button,'String','Cannot plot Stress')
   % Give the edit text box focus so user can correct the error
```

```
    uicontrol(hObject)
end
% --- Executes during object creation, after setting all properties.
function stress_input_CreateFcn(hObject, eventdata, handles)
% hObject handle to stress_input (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
function t_input_Callback(hObject, eventdata, handles)
% hObject handle to t_input (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 t_input as text
%        str2double(get(hObject,'String')) returns contents of t_input as a double
% --- Executes during object creation, after setting all properties.
set(handles.plot_button,'Enable','off')
try
    t = eval(get(handles.t_input,'String'));
    if ~isnumeric(t)
        % t_input is not a number
        set(handles.plot_button,'String','t is not numeric')
    elseif length(t) < 2
        % t_input is not a vector
        set(handles.plot_button,'String','t must be vector')
    elseif length(t) > 1000
        % t_input is too long a vector to plot clearly
        set(handles.plot_button,'String','t is too long')
    elseif min(diff(t)) < 0
        % t_input is not monotonically increasing
        set(handles.plot_button,'String','t must increase')
    else
        % All OK; Enable the Plot button with its original name
        set(handles.plot_button,'String','Plot')
        set(handles.plot_button,'Enable','on')
        return
   end
   % Found an input error other than a bad expression
   % Give the edit text box focus so user can correct the error
   uicontrol(hObject)
 catch EM
    % Cannot evaluate expression user typed
    set(handles.plot_button,'String','Cannot plot t')
    % Give the edit text box focus so user can correct the error
    uicontrol(hObject)
end
% hObject    handle to t_input (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 t_input as text
%        str2double(get(hObject,'String')) returns contents of t_input as a double
% --- Executes during object creation, after setting all properties.
% --- Executes during object creation, after setting all properties.
function t_input_CreateFcn(hObject, eventdata, handles)
% hObject    handle to t_input (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
function initialize_gui(fig_handle, handles,isreset)
% If the metricdata field is present and the reset flag is false, it means
% we are just re-initializing a GUI by calling it from the cmd line
% while it is up. So, bail out as we dont want to reset the data.
if isfield(handles, 'metricdata') && ~isreset
    return;
end
set(handles.datan,'String','');
set(handles.datay,'String','');
set(handles.dataz,'String','');
set(handles.dataj,'String','');
set(handles.datak,'String','');
set(handles.datal,'String','');
set(handles.datam,'String','');
% --- Executes on button press in pushbutton1.
% --- 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)
Irrigation = eval(get(handles.iri_input,'String'));
t = eval(get(handles.t_input,'String'));
stress = eval(get(handles.stress_input,'String'));
i = 1:4;
ETreductionnn = t(i)-t(1);
ETreductionn = t(1);
ETreduction = ETreductionnn./ ETreductionn;
YIELDreductionnn = stress(i)-stress(1);
YIELDreductionn = stress(1);
YIELDreduction = YIELDreductionnn./YIELDreductionn;
EWPPP = (YIELDreduction./ETreduction);
% To calculate Linear yield
z =[ones(size(t)) t];
slopeint =z\stress;
Q = slopeint(1) + (slopeint(2)*t);
```

```
% To calculate response factor
m =[zeros(size(ETreduction)) ETreduction];
slopeinnt = m\YIELDreduction;
slopeinnnt = slopeinnt(2)*1;
qqq = slopeinnt(2)*ETreduction;
% To calculate EWP
ww = slopeint(2)* (t);
EWP = ww./Q;
%To calculate quadratic yield
p = polyfit(t,stress, 2); % Degree 2 fit
y = t*p(2);
z = t.^2*p(1);
x = z + y + p(3);
% To calculate maximum Yield
w =interp1(EWP,stress,slopeinnt(2));
% To calculate maximum ET
B =interp1(EWP,t,slopeinnt(2));
% To calculate maximum Irrigation depth
C = interp1(EWP,Irrigation,slopeinnt(2));
% Create response axes plot in proper axes
plot(handles.Response_axes,ETreduction,qqq,'-*')
set(handles.Response_axes,'XMinorTick','on')
title('Yield Response')
grid on
legend('(1-Ym/Ya) vs (1-ETm/ETa)')
% hObject handle to buttondata_1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
hold off
initialize_gui(gcbf, handles, true);
set(handles.datan,'String',Q');
set(handles.datay,'String',x');
set(handles.datam,'String',EWP');
set(handles.datak,'String',slopeinnnt');
set(handles.dataj,'String',w');
set(handles.datal,'String',C');
set(handles.dataz,'String',B');
```

Peer-review history:
The peer review history for this paper can be accessed here:
http://www.sciencedomain.org/review-history.php?iid=924&id=37&aid=8875