doi: 10.4149/gpb_2024016

A programming toolbox for calculating beta-Euler shape exponents from plant growth data

Jerzy Kosek[1,*] and Mariusz Pietruszka[2,*]

[1] *Liceum Ogólnokształcące KTK, Bielsko-Biała, Poland*
[2] *University of Silesia, Faculty of Natural Sciences, Institute of Biology, Biotechnology and Environmental Protection, Katowice, Poland*

Abstract. Since the acid growth theory was introduced in plant physiology and mainframe computers became more widely available in the mid-20th century, there has been a growing need to accurately predict plant cell morphological parameters during growth. This article presents a computer program that uses an original numerical method to solve a highly nonlinear growth equation. The program is written in Python, a popular open-source scientific software environment called CoCalc or SAGE. This program can be used to determine the growth of an individual plant cell or multicellular organ, such as a coleoptile or hypocotyl segment, at the non-meristemic limit. This standalone program is designed to be user-friendly and accessible to all readers, without barriers. With only a few key parameters, including pH and temperature, this program provides a practical set of tools for comparing growth-related experimental data across various areas of plant biology. Additionally, it could be useful in predicting plant growth during assisted migration, particularly in the face of climate change.

Key words: Assisted migration — Coleoptile — Hypocotyl — pH — Plant cell — Temperature

Introduction

Acid growth involves two acidification parameters: vacuole acidification for tonoplast loosening and cell wall-plasma membrane acidification for both loosening. Additionally, acid growth may require the addition of new components to membranes/cell wall. pH is an essential parameter used to measure proton concentration in natural environments. For living organisms like plant cells and organs, maintaining acid-based homeostasis is crucial to ensure proper physiological functions and growth. Another critical parameter that affects plant growth is the ambient temperature. In the previous article (Pietruszka 2020), we introduced plant growth equations that consider growth's dependence on pH and temperature. We also mentioned the development of a computer program that will allow experimental data to predict plant cell or tissue growth under various treatments. We now keep our promise by providing this program to readers (see Supplementary material).

Potential applications of the presented formalism lie in the α and β parameters (beta-Euler shape exponents) and the direct use of pH(t) experimental data to calculate the cell/organ increase in the volume dV. The latter can be accomplished for experimental control conditions, using biotic/abiotic factors (and mutants), or considered with microscopic theory (Pietruszka 2021). The solution of Eq. (20) in Pietruszka (2020) yielding α and β shape exponents belongs to the highly non-linear optimization problems class. We suppose the biological meaning of the α and β exponents may acquire a connotation for larger biological data sets. Nonetheless, it seems that Eq. (21), *ibid.*, can be treated as the

* The authors equally contributed to the manuscript.
Electronic Supplementary material. The online version of this article (doi: 10.4149/gpb_2024016) contains Supplementary material.
Correspondence to: Mariusz Pietruszka, University of Silesia, Faculty of Natural Sciences, Institute of Biology, Biotechnology and Environmental Protection, Jagiellońska 28, PL-40032 Katowice, Poland
E-mail: mariusz.pietruszka@us.edu.pl

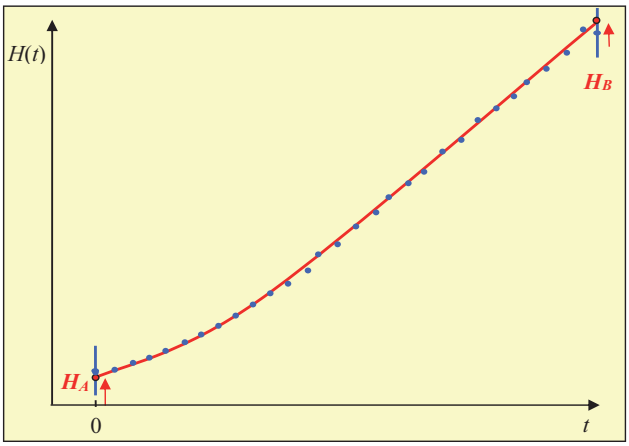

Figure 1. The optimum curve search method. The plant segment growth (height) curve $H(t)$ at a constant temperature, Eq. (4), is given by α and β and the points H_A and H_B; by moving iteratively, H_A and H_B, we find the curve that is best suited to the experimental data (blue points). Parameter c is determined from the values of α, β, H_A, and H_B; see Eq. (8). (See online version for color figure.)

transformation f(pH(t), P, T; α, β) = V(t) for the increasing volume of a plant cell that corresponds to the pH-driven cell enlargement at temperature T.

We refer the reader to recent papers (Pietruszka 2020, 2021) for further experimental and theoretical context on the presented computer program.

Results and Discussion

In the article of Pietruszka (2020), we presented a formalism to describe the process of the increasing volume (V) of a segment of the plant using an augmented form of the Lockhart (1965) equation

$$\frac{1}{V(t)}\frac{dV(t)}{dt} = \Phi\ (P - P_c) \tag{1}$$

where

$$\Phi = \Phi(\mathrm{pH}, T) = c \times T^{\alpha-1}(1-T)^{\beta-1}\,\mathrm{pH}^{\beta-1}\left(1-\mathrm{pH}\right)^{\alpha-1}\Phi_0 \tag{2}$$

as a new method to estimate the Lockhart constant. The exponents α and β and the parameter c are the numbers independent of pH and T, whereas pH(t) and $T(t)$ generally depend on time. In Eq. (1), $P - P_c$ is a constant pressure (P denotes a constant turgor pressure, while P_c denotes the yield threshold), and Φ_0 is the Lockhart constant in $\mathrm{MPa}^{-1}\mathrm{s}^{-1}$. Φ_0 we determine by adjusting the initial experimental data V_{dat} using the Lockhart curve. Applications of this formalism include determining the α, β, and c parameters and directly using pH(t) data to calculate the volume $V(t)$.

In isothermal conditions of growth, we get (see Eq. 21 in Pietruszka 2020)

$$V_{\alpha\beta c,T}(t) = V_{\alpha\beta c,T}(0) \times \exp\left[c\ \Phi_0 (P-P_c) T^{\alpha-1}(1-T)^{\beta-1} \int_0^t \mathrm{pH}_T(\tau)^{\beta-1}\left(1-\mathrm{pH}_T(\tau)\right)^{\alpha-1} d\tau \right] \tag{3}$$

where $V_{\alpha\beta c,T}(t)$ denotes the theoretical volume of a growing segment at constant temperature T (in °C) at time t, $V(0)$ the volume at time $t = 0$ at which growth starts and $\mathrm{pH}_T(t)$ – pH at time t and constant temperature T. The temperature selection is not arbitrary (for example, an optimal growth temperature of T = 25°C in Figure 1 was selected to determine the isothermal curve of maize growth presented in Pietruszka 2020). Temperature T in Eq. (3) is normalized, i.e., divided by T_{max} – the maximum physiological temperature and pH is normalized – by 14.

Assuming that the cross-section of a plant segment is constant during growth, we have an analogous relationship for the length $H(t)$ of a segment

$$H_{\alpha\beta c,T}(t) = H_A \times \exp\left[c\ \Phi_0 (P-P_c) T^{\alpha-1}(1-T)^{\beta-1} \int_0^t \mathrm{pH}_T(\tau)^{\beta-1}\left(1-\mathrm{pH}_T(\tau)\right)^{\alpha-1} d\tau \right] \tag{4}$$

where $H_A \equiv H(0)$ – initial length.

The expression in the exponent of equations (3) and (4) of the form

$$T^{\alpha-1}(1-T)^{\beta-1}\int_0^t \mathrm{pH}(\tau,T)^{\beta-1}\left(1-\mathrm{pH}(\tau,T)\right)^{\alpha-1} d\tau \tag{5}$$

depends on the temperature T – by changing T, we change the characteristics of $\mathrm{pH}_T(t)$, which changes the value of the integral, so T will change the whole exponent. It follows that the increase depends on the temperature and that there is an optimum temperature T_{opt} at which growth is the fastest.

Determining parameters α, β, and c – the algorithm

Parameters α, β, and c for a given plant can be determined in the following way. During growth at constant temperature T, the length $H_{dat}(t_j)$ of a plant coleoptile segment and pH(t_j) of the incubation medium is measured; $t_j = Dt \times j$, $j = 0, \ldots, n-1$ times of measurements, while Dt denotes the time interval between measurements. For a detailed description of the experimental method, see Pietruszka (2020).

The problem of finding parameters α, β, and c is simplified by approximating the measurement data with curve $H_{\alpha\beta c,T}(t)$. We impose the following conditions:

1. For all points from the $H_{dat}(t_j)$, $j = 0,..., n-1$ measurement, the deviation of the curve from the measurement points should not exceed the measurement uncertainty DH, i.e.

$$\bigvee_{j=0,n-1} \left| H_{\alpha\beta c,T}(t_j) - H_{dat}(t_j) \right| \le DH \tag{6}$$

2. The $H_{\alpha\beta c,T}(t)$ curve approximates the experimental data with a square deviation Dev($\alpha\beta c$) that is no greater than the maximum allowable deviation

$$\mathrm{Dev}(\alpha\beta c) = \sum_{j=n_1}^{n_2} \left(H_{\alpha\beta c,T}(t_j) - H_{dat}(t_j) \right)^2 \le (n_2 - n_1 + 1) \times \times DH^2 \tag{7}$$

where $H_{\alpha\beta c,T}(t_j)$, is the length of a plant segment at time t_j calculated from Eq. (4), $H_{dat}(t_j)$ – length of a plant segment measured at time t_j and n_1, n_2 – the indices are such that pH(t_j) decreases for $t_{n1} < t_j < t_{n2}$.

The problem of minimizing the above equations belongs to the class of highly non-linear optimization problems. This equation has elements in the form exp[$f(\alpha, \beta, H_A, H_B)$ × integral (α, β)], and several local extremes also exist. It can be simplified as explained below.

Determining parameter c depending on α, β and the two terminal points of the curve

We note that the curve $H_{\alpha\beta c,T}(t)$, which is given by Eq. (4), is clearly defined by four parameters: α, β, c, and the starting point $H_A \equiv H(0)$. The numerical value of parameter c is many orders of magnitude greater than α and β and is difficult to estimate initially. Therefore, it is more convenient to eliminate parameter c at the beginning of the calculations. Assuming that the curve passes through two points – $H_A \equiv H_A(t)$ and $H_B \equiv H_B(t)$ from Eq. (4), we get

$$c = \frac{\ln(H_B / H_A)}{f(\alpha,\beta)\, g(\alpha,\beta,t)} \tag{8}$$

where: $f(\alpha,\beta) = \Phi_0 (P - P_c)\, T^{\alpha-1}(1-T)^{\beta-1}$ and

$$g(\alpha,\beta,t) = \int_0^t \mathrm{pH}_T(\tau)^{\beta-1} \left(1 - \mathrm{pH}_T(\tau)\right)^{\alpha-1} d\tau$$

Substituting the expression Eq. (8) to equation Eq. (4), we get an equation in which the new H_B parameter replaces c.

The iterative method of searching for the H_A, H_B, and α, β parameters is as follows:

1. Take the initial values $\alpha = 0$, $\beta = 0$.
2. Take the initial values $H_A = H_{dat}(0) - DH$, $H_B = H_{\mathrm{dat}}(n-1) - DH$.
3. Determine whether the conditions for a good approximation are met, Eqs. (6) and (7); write down any good solutions.
4. Increase the H_A and H_B values with a step equal to DH/10 until $H_A = H_{dat}(0) + DH$, $H_B = H_{\mathrm{dat}}(n-1) + DH$ is reached (this corresponds to the displacement of the H_A and H_B points along the points of the measurement uncertainty) and determine whether the conditions (6) and (7), etc. are met again.
5. By doing so, we find the best-fit curve $H_{\alpha\beta c,T}(t)$ for the experimental data, corresponding to the smallest square deviation $Dev(\alpha\beta c)$; parameter c is then calculated from Eq. (8).
6. Iteratively change α and β in the range (0, αβ_max) and repeat the process of moving the H_A and H_B points – points 2–5 of this method.

Finally, we find a set of possible numbers $\{\alpha, \beta, c, H_A\}$ that defines the plant growth curves fitted to experimental data.

Constants calculation results

The data from the segment height and pH measurement described in Pietruszka (2020) were recorded in the vectors $\mathrm{H_{dat}}[j]$, pH [j], and j = [0, 32]. The $\{\alpha, \beta, c, H_A\}$ parameters were calculated as follows: the values of α and β were changed in the range (0, 50), and the starting and end points H_A and H_B of the growth curve were changed in the steps of DH/10 = 0.001 cm, where

1. The H_A point was shifted in the rectangle of the uncertainty of the H_{dat} point measurement [0] in the range ($H_{\mathrm{dat}}[0] - DH$, $H_{\mathrm{dat}}[0] + DH$).
2. The H_B point was shifted in the rectangle of the uncertainty of the H_{dat} point measurement [n–1] in the range ($H_{\mathrm{dat}}[n-1] - DH$, $H_{\mathrm{dat}}[n-1] + DH$).

The growth curve was calculated for the given numbers $\{\alpha, \beta, H_A, H_B\}$, and it was determined whether the conditions of a good approximation were met

$$\left| H_{dat}[j] - H_{\alpha\beta}[j] \right| \le DH,\ j = 0, n-1 \tag{9}$$

i.e., the relevant points of the theoretical curve fit within the rectangles of the uncertainty of measured points.

An extensive set of possible parameter values (α and β) that met the above criterion was obtained from the calculations. The points that were found are presented in Figure 2 (see the caption). Figure 3 shows the values of the most minor square deviations that were found, depending on the α parameter, for the different values of the β parameter

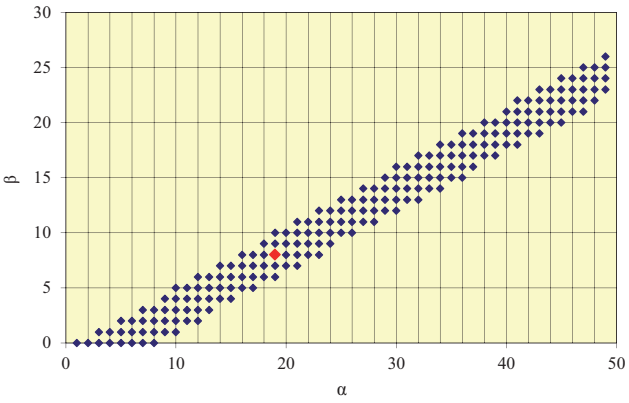

Figure 2. Points (α, β) for which the $H(j)$ curve meets the criterion of good approximation Eq. (6), i.e., the deviations of the growth curve that are measured relative to all of the measurement points are smaller than the measurement uncertainty (error) DH = 0.01 cm. The point (α = 19, β = 8), found in the program's original version, is marked in red where instead of an iterative shifting of H_A, and H_B, the c parameter was iteratively changed, and $H_A = H_{dat}$ [0] = const. Note that the experimental (input) data must reach the growth saturation area to reduce the result's uncertainty. (See online version for color figure.)

(and H_A and H_B, respectively, for which the total deviations $Dev(\alpha, \beta, H_A, H_B)$ were the lowest). It can be seen that for a given value of α, there was an optimal parameter β for which the deviation was the smallest. The deviation that was found in the previous version of the program is marked in red. It can be seen that there is no preferential pair of numbers (α and β) for which the deviation would be the smallest.

It turned out there is a set of theoretical H curves (α, β, H_A, H_B), which correctly approximate the experimental data, while the acceptable values of α and β change in an extensive range (however, the least values of α and β can be accepted). We assume that a narrower selection of the α and β parameter values will be possible when $H_{dat}(t)$ and pH(t) measurements are taken over a longer time during which the plant growth rate is significantly reduced (the saturation end of the sigmoid growth curve). This is the same problem as was already reported by Zajdel et al. (2016). Both complementary methods (programs) that can run as stand-alone applications can be used directly from the attached links.

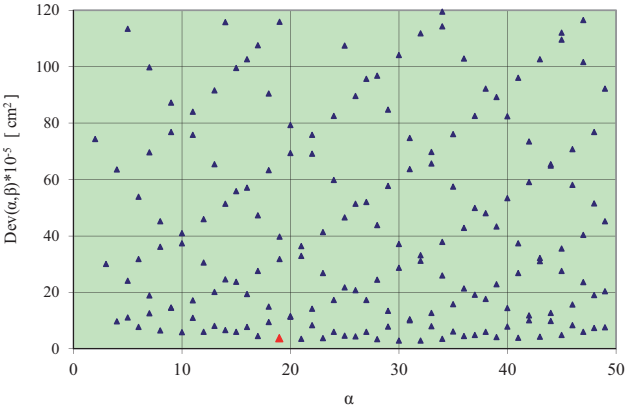

Figure 3. The values of the minimum square deviations $Dev(\alpha, \beta)$ depending on parameter α for different values of parameter β. The point, found in the program's original version, is marked in red. (See online version for color figure.)

We have developed a programming toolbox to calculate beta-Euler shape exponents based on plant growth data. As an example that shows the possibilities of Eq. (3), note that this formula may be embedded in an evolutionary context, which is related to the migration of plants away from the equator (with changes in latitudes) as climate change as well as their adaptation to the spatial distribution of pH in the soil as a substitute for a high temperature (see "Migration of Plants" in Pietruszka 2020, or more recent article of Pietruszka 2023). In the era of climate change and ongoing attempts at the assisted migration of plant species or other analytical (e.g., comparative) studies, the potential use of the program that we presented here, which can be copied and pasted into any Python environment [SAGE CoCalc (Collaborative Calculation and Data Science); https://cocalc.com or any other Python/SAGE environment], may prove helpful among plant biologists and ecologists.

Conflict of interest. There is no conflict of interest.

Appendix

Initial data in the computer program

Initial data can be introduced directly into the program at the DATA part simply by substituting the exemplary data we have taken from the experiment. They include the following:

1. n – number of measurements of pH and H_dat during the growth of a plant
2. Dt – the interval between the measurements in seconds [s]
3. T – temperature on the Celsius scale [°C] (there is no need to normalize it – the computer program will normalize it)
4. T_max – the maximum physiological temperature of a plant on the Celsius scale [°C]
5. pH = vector(RR, n) – the vector of the n measured values of pH, in [0, 14] (there is no need to normalize it – the computer program will normalize it)
6. H_dat = vector(RR, n) – the vector of the n measured lengths of a coleoptile segment in [cm]

7. DH – the accuracy of the H_{dat} measurements in [cm]
8. F_0 – the Lockhart normalising constant in [1/(MPa*s)]. It is such that the Lockhart curve, i.e., the solution of the Lockhart equation approximates the initial part of data H_dat; e.g. F_0 = 10^(−5) = Φ_0 well in Pietruszka (2020); PY = P – Pc in [MPa] where P – turgor pressure, Pc – yield threshold, both in [MPa].

References

Lockhart A (1965): An analysis of irreversible plant cell elongation. J. Theor. Biol. **8,** 264-27510 https://doi.org/10.1016/0022-5193(65)90077-9

Pietruszka MA (2020): Chemical potential-induced wall state transitions in plant cell growth. J. Plant Growth Regul. **39,** 841-866 https://doi.org/10.1007/s00344-019-10026-x

Pietruszka MA (2021): Expansive growth vs. pH reflects a Poisson point process of binding/unbinding events in plant cell walls. J. Plant Growth Regul. **41,** 2543-2554 https://doi.org/10.1007/s00344-021-10459-3

Pietruszka MA (2023): Collective excitations of germinating pollen grains at critical points. Sci. Rep. **13,** 610 https://doi.org/10.1038/s41598-023-27754-6

Zajdel P, Pietruszka M, Haduch-Sendecka A (2016): Application of the effective formula of growth functional to quantitative description of growth of plant cells. Acta Physiol. Plant **38,** 1-9 https://doi.org/10.1007/s11738-016-2233-4

Received: January 2, 2024
Final version accepted: April 21, 2024

doi: 10.4149/gpb_2024016

Supplementary Material

A programming toolbox for calculating beta-Euler shape exponents from plant growth data

Jerzy Kosek[1,*] and Mariusz Pietruszka[2,*]

[1] *Liceum Ogólnokształcące KTK, Bielsko-Biała, Poland*
[2] *University of Silesia, Faculty of Natural Sciences, Institute of Biology, Biotechnology and Environmental Protection, Katowice, Poland*

Supplementary Information

```
# SAGE PROGRAM: Plant segment height parameterized by pH and T
# Execution takes several minutes on a PC
#
n    = 33                 # n – number of pH[] and H_dat[] data; change for your own data
pH  = vector(RR,n)        # measured pH; pH will be scaled [0, 14] →[0, 1]
pH_sin = vector(RR,n)  # auxiliary
# insert your own experimental data
pH[ 0]  =5.674; pH[ 1]  =5.737; pH[ 2]  =5.782; pH[ 3]  =5.791; pH[ 4]  =5.761;
pH[ 5]  =5.695; pH[ 6]  =5.598; pH[ 7]  =5.469; pH[ 8]  =5.323; pH[ 9]  =5.181;
pH[10] =5.063; pH[11] =4.928; pH[12] =4.775; pH[13] =4.666; pH[14] =4.604;
pH[15] =4.566; pH[16] =4.523; pH[17] =4.502; pH[18] =4.462; pH[19] =4.438;
pH[20] =4.420; pH[21] =4.405; pH[22] =4.390; pH[23] =4.377; pH[24] =4.376;
pH[25] =4.366; pH[26] =4.356; pH[27] =4.347; pH[28] =4.339; pH[29] =4.330;
pH[30] =4.324; pH[31] =4.323; pH[32] =4.322;

H_dat=vector(RR,n)  # H_dat[j], in [cm], plant segment length measured at time t_j=Dt*j
# insert your experimental data
H_dat[ 0] =1.0;         H_dat[ 1]=1.00356;  H_dat[ 2]=1.00508;   H_dat[ 3]=1.00711;
H_dat[ 4] =1.00965;  H_dat[ 5]=1.01220;  H_dat[ 6]=1.01474;   H_dat[ 7]=1.01778;
H_dat[ 8] =1.02185;  H_dat[ 9]=1.02540;  H_dat[10]=1.02998;  H_dat[11]=1.03455;
H_dat[12]=1.03963;  H_dat[13]=1.04472;  H_dat[14]=1.05030;  H_dat[15]=1.05640;
H_dat[16]=1.06200;  H_dat[17]=1.06809;  H_dat[18]=1.07520;  H_dat[19]=1.08181;
H_dat[20]=1.08841;  H_dat[21]=1.09553;  H_dat[22]=1.10213;  H_dat[23]=1.10925;
H_dat[24]=1.11585;  H_dat[25]=1.12246;  H_dat[26]=1.12957;  H_dat[27]=1.13567;
H_dat[28]=1.14177;  H_dat[29]=1.14787;  H_dat[30]=1.15447;  H_dat[31]=1.16006;
H_dat[32]=1.16616;

DH  =  0.01       # DH – measurement uncertainty of H_dat, in [cm]
T  = 25           # T – temperature, in [°C]; it will be scaled [0,T_max]°C →[0,1]; insert temp.
T_max = 45        # T_max, in [°C] – maximal physiological temperature
Dt    = 900       # Dt – interval between measurements, in [s]
PY    =  0.3      # PY = P–Pc, in [MPa], P-turgor pressure, Pc-yield threshold, both in [MPa]
F_0   = 10^(-5)  # F_0 – Lockhart constant, in [1/(MPa s)]

for j in range(n):
    pH[j]=pH[j]/14.0
```

```
p1=point([(j, pH[j]) for j in range(n)] )
p1.axes_labels([“Data point index”,”pH scaled to [0, 1]”])
p1.show(frame=True, figsize=5)
print “Fig. 1 pH[j], j = 0,”,n-1 , “scaled [0, 14] → [0, 1]”

p=point([(j, H_dat[j]) for j in range(n)] )
p.axes_labels([“Data point index”,’Segment length H_dat [cm]’])
p.show(frame=True, figsize=5)
print “Fig. 2 H_dat[j], j = 0,”,n-1
print
T=T/T_max
print “Temperature was scaled [0, T_max = “, T_max,”]°C → [0, 1]”
print
print “END OF INPUT DATA MODULE. RUN CALCULATIONS IN THE NEXT CELL”

import numpy as np
import numpy.linalg

# xx – measured data index vector
xx=np.array([0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,\
                    26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,1,42,43,44,45,46,47,48,\
                    49,50,51,52,53,54,55,56,57,58,59,60])

var (‘v’)
H_MAP = vector(RR,n) # H_MAP[j] – segment hight at t_j = Dt*j
W_Dat = list_plot(list(zip(xx, H_dat)), color=’blue’,  figsize=5.5, frame=’true’, legend_label=”data”)
W_Loc=plot(H_dat[0]*exp(PY*F_0* Dt*v),(v,0,n), color=’brown’, thickness=1, linestyle=’:’, legend_label=”Lockhart”)

Dev_H_min_abs  = 10^8  # do not remove

for aa in range(0, 50):       # XXXX – numbers aa≡alpha in range(0, 50) can be freely changed (integer)
    for bb in range(0, 28):  # XXXX – numbers bb≡betha in range(0, 28) can be freely changed (integer)
            #print “alpha= %6.3f,   beta= %6.3f” % (aa,bb)

            Dev_H_min = 10^8     # do not remove
            Pointer_ab = False
            Calculation of the constant factor calka
            calka=0
            s_0= pH[0]^(bb-1)*(1-pH[0])^(aa-1)

    for j in range(0,n-1):
            s_1 = pH[j+1]^(bb-1)*(1-pH[j+1])^(aa-1)
            calka=calka+ Dt/2*(s_0 + s_1)
            s_0=s_1      # OK

       # A. Change of points H_A and H_B of the curve
       for i_A in range(-10,11):
          for i_B in range(-10,11):
             H_A = H_dat[0]   + DH/10 * i_A  #change of the starting point
             H_B = H_dat[n-1] + DH/10 * i_B  #change of the endpoint
             cc = ln(H_B/H_A)/(T^(aa-1) * (1-T)^(bb-1) * F_0 * PY * calka)    # cc(aa,bb) calculation
             pom_k = T^(aa-1) * (1-T)^(bb-1) * cc * F_0 * PY
             s_0= pH[0]^(bb-1)*(1-pH[0])^(aa-1)
```

```
            # B. Calculation of the growth curve H_MAP
            # C. The pointer “Pointer” changes to False if at any point abs(H_MAP[j+1]-H_dat[j+1]) > DH,
            # i.e the curve deviation from H_dat[j] exceeds the uncertainty DH of H_dat measurement;
            # such solutions are ignored
            # D. Calculation of deviation Dev
            Pointer = True

            H_MAP[0] = H_A   # initial point of theoretical curve

            Dev_H = (H_MAP[0]-H_dat[0])^2
            s_0= pH[0]^(bb-1)*(1-pH[0])^(aa-1)
            for j in range(0,n-1): #range(n_1,n_2-1):
               s_1 = pH[j+1]^(bb-1)*(1-pH[j+1])^(aa-1)
               H_MAP[j+1]=H_MAP[j]*exp(pom_k*Dt/2* (s_0 + s_1))
               Dev_H = Dev_H + (H_MAP[j+1]-H_dat[j+1])^2
               s_0=s_1
               if abs(H_MAP[j+1]-H_dat[j+1]) > DH:
                  Pointer = False

            if Pointer == True and Dev_H < Dev_H_min: # record of the solution
               Pointer_ab = True
               Dev_H_min = Dev_H
               #print “H_A= %6.3f,   H_B= %6.3f” % (H_A, H_B)
               print “alpha= %6.3f,   beta= %6.3f,   Dev_H_min= %8.7f” % (aa,bb, Dev_H_min)

               #W_MAP=list_plot(H_MAP, plotjoined=True,color=’red’, legend_label=”abc curve”)
               #p=W_Dat+W_MAP
               #p.show(figsize=3)

               # Record of the best minimum
               if Dev_H_min < Dev_H_min_abs:
                  Dev_H_min_abs = Dev_H_min
                  aa_min_abs=aa
                  bb_min_abs=bb
                  cc_min_abs=cc
                  H_A_min_abs=H_A
                  H_B_min_abs=H_B

                  print “alpha= %6.3f,   beta= %6.3f,   Dev_H_min_abs= %8.7f” %\
                  (aa_min_abs,bb_min_abs, Dev_H_min_abs)

                  W_MAP_min_abs=list_plot(H_MAP, plotjoined=True,color=’red’,legend_label=”abc curve”)
                  p=W_Dat+W_MAP_min_abs
                  p.show(figsize=3)

      if Pointer_ab == False: print “alpha= %6.3f,   beta= %6.3f     – no solution” % (aa,bb)

# Final print
print
print “SEARCH RESULTS”
print “alpha= %7.6f,   beta= %7.6f,   c= %8.5e  Dev_H_min_abs =%12.11f”%\
(aa_min_abs, bb_min_abs, cc_min_abs, Dev_H_min_abs)
#print  “H_dat[0]  = %6.3f,     H_dat[n-1] = %6.3f” % (H_dat[0], H_dat[n-1])
```

```
print  “H_A        = %6.3f,     H_B          = %6.3f” % (H_A_min_abs, H_B_min_abs)
p=W_Dat+W_Loc+W_MAP_min_abs
p.axes_labels([“Data point index”,’Plant segment height [cm]’])
p.show(figsize=6)
```