*Appendix C*

# SOLUTIONS TO VERIFICATION PROBLEMS IN CHAPTER 3

## C.1. INTRODUCTION

This appendix contains the analytical solutions for the three verification problems discussed in Chapter 3.

## C.2. VERIFICATION PROBLEM ONE – TERZAGHI ONE-DIMENSIONAL CONSOLIDATION

Terzaghi (1925) derived the following partial differential equation to describe the settlement of a saturated clay layer and dissipation of excess pore pressures with time:

$$c_v \frac{\partial^2 u}{\partial^2 z} = \frac{\partial u}{\partial t} \qquad \text{Equation C.40}$$

Where,

$c_v$ is the coefficient of consolidation
$u$ is the excess pore water pressure in the soil
$t$ is time
$z$ is depth

The boundary and initial conditions are

$$u(z,0) = q \qquad\qquad 0 \le z \le 2H$$
$$u(0,t) = 0 \qquad u(2H,t) = 0 \qquad \text{for } t > 0$$

H is the length of the longest drainage path. The separation of variables technique is used to obtain a solution to the partial differential equation for the prescribed boundary and initial conditions. The resulting solution for $u(z,t)$ is a convergent infinite series:

$$\frac{u}{q}=\sum_{n=0}^{\infty}\frac{4}{(2n+1)\pi}\sin\left(\frac{(2n+2)\pi}{2}\frac{z}{H}\right)\exp\left(-\frac{(2n+1)^2\pi^2}{4}T\right)$$ Equation C.41

Where

$T$ is the dimensionless time factor $\left(=\frac{c_v t}{H^2}\right)$

The degree of consolidation at any depth z is given by $U_z=1-\frac{u}{q}$. The average degree of consolidation, U, is obtained by integrating the degree of consolidation, $U_z$, over the entire soil column and dividing by the height of the soil column.

$$U=1-\sum_{n=0}^{\infty}\frac{8}{(2n+1)^2\pi^2}\exp\left[\frac{(2n+1)^2\pi^2}{4}T\right]$$ Equation C.42

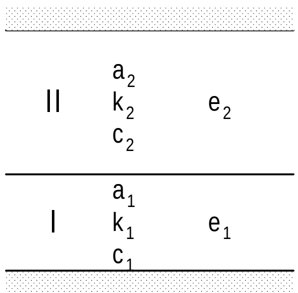
The expressions for $\frac{u}{q}$ and $U$ are easy to evaluate using either a programmable calculator or a spreadsheet.

## C.3. VERIFICATION PROBLEM TWO – CONSOLIDATION OF CONTIGUOUS CLAY LAYERS WITH DIFFERENT PERMEABILITY

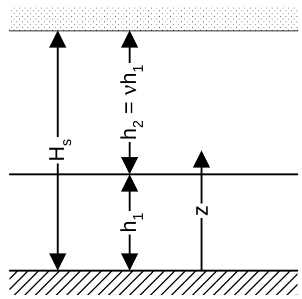
Gray (1944) developed analytical solutions to general one-dimensional problem of two consolidating adjacent compressible strata. Gray's solution and its numerical evaluation for verification problem 2 are demonstrated in the following Mathcad® worksheet, 1-D contiguous layers.mcd.

**The purpose of this Mathcad worksheet is to evaluate the expressions for the excess pore pressure in two contiguous clay of unlike compressibility developed by** *Gray, H. (1944), "Simultaneous consolidation of contiguous layers of unlike compressible soils", ASCE Transactions, No. 2258, pp. 1327-1356.*

## The general cases of two adjacent compressible strata:




**Case a**. Free drainage at top and bottom




**Case b**. Free drainage at top only

### *Terminology and definitions*

k = coefficient of permeability
$a_v$ = coefficient of compressibility
e = void ratio
$e_0$ = initial void ratio
c = coefficient of consolidation
p = surface traction applied at t = 0

$$H_s = \frac{H}{1+e} \qquad z = \frac{h}{1+e} \qquad k_s = \frac{k}{1+e}$$

$$c = \frac{k}{a_v \cdot \gamma \cdot (1+e)}$$

### *stress-strain relations:*

the soil is assumed to be linearly elastic

$\mu$ = poisson's ratio

E = Young's modulus

D = constrained modulus

$m_v$ = coefficient of compressibility

$$D = \frac{E \cdot (1-\mu)}{(1+\mu) \cdot (1-2 \cdot \mu)} \qquad G = \frac{E}{2 \cdot (1+\mu)}$$

$$m_v = \frac{1}{D} \qquad a_v = \frac{m_v}{(1+e_0)}$$

Define four dimensionless numbers ($\mu$, $\sigma$, $\nu$, T):

$$\mu^2 = \frac{c_1}{c_2} \qquad \sigma = \frac{1}{\mu} \cdot \frac{k_{s1}}{k_{s2}} \qquad \nu = \frac{h_2}{h_1} \qquad T = \frac{c_1}{(h_1)^2} \cdot t$$

**Note**: The time factor, T, is based only on the properties of layer I and the time, t.

$u_1$ = excess pore pressure in layer I
$u_2$ = excess pore pressure in layer II
$U_1$ = average degree of consolidation in layer I
$U_2$ = average degree of consolidation in layer II

Gray (1944) developed the following analytical solutions for the two cases:

## Case a

$$u_1 = \sum_{n=1}^{\infty} C_n e^{-TA_n^2} \sin( A_n \frac{z}{h_1}) \sin( \mu\nu A_n )$$

$$u_2 = \sum_{n=1}^{\infty} C_n e^{-TA_n^2} \sin( A_n ) \sin( 1 + \nu - \frac{z}{h_1}) \mu A_n$$

$$U_1 = 1 - 2\sum_{n=1}^{\infty} \frac{\sin(\mu\nu A_n)(\sigma \sin(\mu\nu A_n) + \sin A_n)}{(A_n)^2 (\sigma \sin^2 (\mu\nu A_n) + \mu\nu \sin^2 (A_n))} (1 - \cos A_n) e^{-TA_n^2}$$

$$U_2 = 1 - \frac{2}{\mu\nu} \sum_{n=1}^{\infty} \frac{\sin A_n (\sigma \sin \mu\nu A_n + \sin A_n)}{(A_n)^2 (\sigma \sin^2 \mu\nu A_n + \mu\nu \sin^2 A_n)} (1 - \cos \mu\nu A_n) e^{-TA_n^2}$$

in which $A_n$ must be a root of

$$F(A) = \sigma \cos A \sin \mu\nu A + \sin A \cos \mu\nu A$$

and

$$C_n = 2p \frac{\sigma \sin \mu\nu A_n + \sin \mu\nu A_n}{\sigma A_n \sin^2 \mu\nu A_n + \mu\nu \sin^2 A_n}$$

## Case b

$$u_1 = \sum_{n=1}^{\infty} C_n e^{-TA_n^2} \cos( A_n \frac{z}{h_1}) \sin( \mu\nu A_n )$$

$$u_2 = \sum_{n=1}^{\infty} C_n e^{-TA_n^2} \cos( A_n ) \sin( 1 + \nu - \frac{z}{h_1}) \mu A_n$$

$$U_1 = 1 - 2\sum_{n=1}^{\infty} \frac{\sin( A_n )(\cos( A_n ) \sin \mu\nu A_n )}{(A_n)^2 (\sigma \sin^2 (\mu\nu A_n) + \mu\nu \cos^2 (A_n))} e^{-TA_n^2}$$

$$U_2 = 1 - \frac{2}{\mu\nu} \sum_{n=1}^{\infty} \frac{\cos^2 A_n (1 - \cos \mu\nu A_n)}{(A_n)^2 (\sigma \sin^2 \mu\nu A_n + \mu\nu \cos^2 A_n)} e^{-TA_n^2}$$

in which $A_n$ must be a root of

$$G(A) = \sigma \sin A \sin \mu\nu A + \cos A \cos \mu\nu A$$

and

$$C_n = \frac{2p \cos A_n}{\sigma A_n \sin^2 \mu\nu A_n + \mu\nu A_n \cos^2 A_n}$$

Develop solution for Case b using Gray's solution.  First, set up procedure for finding roots of G(A).

$$G(A,\sigma,\mu,\nu) := \sigma\cdot\sin(A)\cdot\sin(\mu\cdot\nu\cdot A) - \cos(A)\cdot\cos(\mu\cdot\nu\cdot A)$$

Define the function **zbrak** which will bracket the first n roots of G(A,σ,μ,ν).  The brackets for each root are returned as a row in an array.  The values of the left and right brackets are in column zero and one, respectively.  The variable inc controls the step size of A that **zbrak** uses when incrementing A.

```
zbrak(n,σ,μ,ν,inc) := | W ← 0
                      | A ← 0
                      | B ← A
                      | fa ← G(A,σ,μ,ν)
                      | for i ∈ 1 .. n
                      |   | check ← 0
                      |   | while check = 0
                      |   |   | B ← B + inc
                      |   |   | fb ← G(B,σ,μ,ν)
                      |   |   | if fa·fb < 0
                      |   |   |   | check ← 1
                      |   |   |   | W[i-1,0] ← A
                      |   |   |   | W[i-1,1] ← B
                      |   |   |   | A ← B
                      |   |   |   | fa ← fb
                      |   |   | if fa·fb > 0
                      |   |   |   | fa ← fb
                      |   |   |   | A ← B
                      | W
```

Define the function **brent**, which uses Brent's method for finding the root of a function known to lie between a and b.  The root is refined until its accuracy is tol or the maximum no. of iterations is reached.

First, define the functions **test** and **test1** which will perform logical tests for **brent**.

```
test(fb,fc) := | if fc>0      if fb>0
               |   | 1
               |   | break
               | if fc<0   if fb<0   otherwise
               |   | 1
               |   | break
               | -1

test1(e,toli,fa,fb) := | 1   if |fa| > |fb|   if |e| ≥ toli
                       | -1  otherwise
```

Machine ε:

$$\varepsilon_{mach} := 1\cdot 10^{-15}$$

Reference for Brent's method: Brent, R.P. 1973, *Algorithms for Minimization without Derivatives* (Englewood Cliffs, NJ: Prentice-Hall), Chapters 3, 4.

```
brent(a, b, σ, μ, ν, tol) :=  itmax ← 100
                              fa ← G(a, σ, μ, ν)
                              fb ← G(b, σ, μ, ν)
                              break  if  fa·fb > 0
                              c ← a
                              fc ← fa
                              d ← b − a
                              e ← d
                              for  i ∈ 1 .. itmax
                                 if  test(fb, fc) > 0
                                    c ← a
                                    fc ← fa
                                    d ← b − a
                                    e ← d
                                 if  |fc| < |fb|
                                    a ← b
                                    b ← c
                                    c ← a
                                    fa ← fb
                                    fb ← fc
                                    fc ← fa
                                 toli ← 2·ε_mach·|b| + 0.5·tol
                                 xm ← 0.5·(c − b)
                                 b
                                 break  if  |xm| ≤ toli
                                 break  if  fb = 0
                                 if  test1(e, toli, fa, fb) > 0
                                    s ← fb/fa
                                    if  a = c
                                       p ← 2·xm·s
                                       q2 ← 1.0 − s
                                    otherwise
                                       q2 ← fa/fc
                                       r ← fb
```

```
r←  fb/fc
p← s·(2·xm·q2·(r − q2) − (b − a)·(1.0 − r))
q2← (q2 − 1.0)·(r − 1.0)·(s − 1.0)
q2← -q2  if  p>0
p← |p|
if 2·p<min([3·xm·q2 − |toli·q2| ; |e·q2|])
    e← d
    d← p/q2
otherwise
    d← xm
    e← d
otherwise
    d← xm
    e← d
a← b
fa← fb
b← b + d  if |d|>toli
b← b + |toli|·|xm|/xm   otherwise
fb← G(b,σ,μ,ν)
b
```

Define remaining functions necessary for solving an example problem for Case b.

$INC := 0.1$ $\quad i := 12$   the variable i will determine how many terms are evaluated when approximating the infinite series.

$$u_1(p,\sigma,\mu,\nu,z,h_1,T) := \left| \begin{array}{l} sum \leftarrow 0 \\ W \leftarrow zbrak(i,\sigma,\mu,\nu,INC) \\ \text{for } n \in 1..i \\ \quad \left| \begin{array}{l} An \leftarrow brent(W_{n-1,0}, W_{n-1,1}, \sigma,\mu,\nu,TOL) \\ Cn \leftarrow \dfrac{2\cdot p\cdot \cos(An)}{\sigma\cdot An\cdot \sin(\mu\cdot\nu\cdot An)^2 + \mu\cdot\nu\cdot An\cdot\cos(An)^2} \\ term \leftarrow Cn\cdot\exp(-T\cdot An^2)\cdot\cos\left(An\cdot\dfrac{z}{h_1}\right)\cdot\sin(\mu\cdot\nu\cdot An) \\ sum \leftarrow sum + term \end{array}\right. \\ sum \end{array}\right.$$

:excess pore pressure in layer I

$$u_2(p,\sigma,\mu,\nu,z,h_1,T) := \left| \begin{array}{l} sum \leftarrow 0 \\ W \leftarrow zbrak(i,\sigma,\mu,\nu,INC) \\ \text{for } n \in 1..i \\ \quad \left| \begin{array}{l} An \leftarrow brent(W_{n-1,0}, W_{n-1,1}, \sigma,\mu,\nu,TOL) \\ Cn \leftarrow \dfrac{2\cdot p\cdot \cos(An)}{\sigma\cdot An\cdot \sin(\mu\cdot\nu\cdot An)^2 + \mu\cdot\nu\cdot An\cdot\cos(An)^2} \\ term \leftarrow Cn\cdot\exp(-T\cdot An^2)\cdot\cos(An)\cdot\sin\left[\mu\cdot An\cdot\left(1+\nu-\dfrac{z}{h_1}\right)\right] \\ sum \leftarrow sum + term \end{array}\right. \\ sum \end{array}\right.$$

:excess pore pressure in layer II

$$u(p,\sigma,\mu,\nu,z,h_1,T) := \left| \begin{array}{ll} u_1(p,\sigma,\mu,\nu,z,h_1,T) & \text{if } z \le h_1 \\ u_2(p,\sigma,\mu,\nu,z,h_1,T) & \text{otherwise} \end{array}\right.$$

:general expression for excess pore pressure at elevation z.

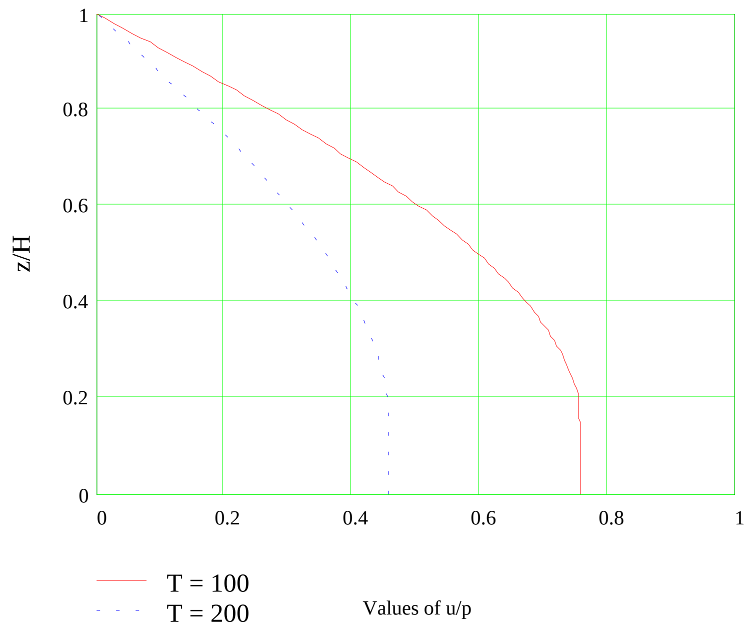
$$Data(p,\sigma,\mu,\nu,h_1,T,inc) := \left| \begin{array}{l} a_{0,0} \leftarrow 0 \\ a_{0,1} \leftarrow u(p,\sigma,\mu,\nu,a_{0,0},h_1,T) \\ step \leftarrow \dfrac{h_1 + \nu\cdot h_1}{inc} \\ \text{for } j \in 1..inc \\ \quad \left| \begin{array}{l} a_{j,0} \leftarrow a_{j-1,0} + step \\ a_{j,1} \leftarrow u(p,\sigma,\mu,\nu,a_{j,0},h_1,T) \end{array}\right. \\ a \end{array}\right.$$

**Data** is a function for generating an excess pore pressure isochrones.

## Example Problem (after Gray, H. 1944)

$p := 100 \quad \sigma := 2 \quad \mu := 5 \quad \nu := 4 \quad h_1 := 2 \quad e_1 = e_2$

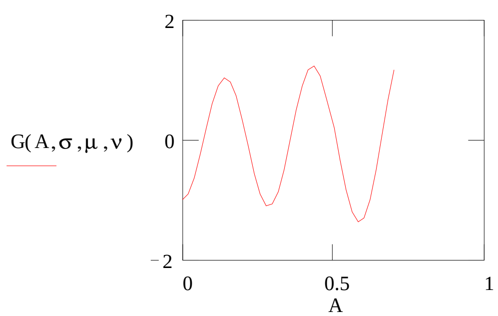
$H := \mathrm{Data}(p, \sigma, \mu, \nu, h_1, 100, 100) \quad I := \mathrm{Data}(p, \sigma, \mu, \nu, h_1, 200, 100)$




The solution for this example agrees with Gray's solution.

Example of how the first i roots of G(A) are found

$A := 0, .02 .. .7$




$i := 5$

First, **zbrak** is called to bracket the roots

$I := \mathrm{zbrak}(i, \sigma, \mu, \nu, .05)$

$j := 1 .. i$

Next, **brent** is called to find the roots.
**Brent** uses the brackets found by **zbrak**.

$r_{j-1} := \mathrm{brent}\left(I_{j-1,0}, I_{j-1,1}, \sigma, \mu, \nu, 1 \cdot 10^{-13}\right)$

$sol_{j-1} := G\left(r_{j-1}, \sigma, \mu, \nu\right)$ check the roots

<u>brackets</u> <u>roots</u> <u>Value of G at roots</u>

$$I = \begin{bmatrix} 0.05 & 0.1 \\ 0.2 & 0.25 \\ 0.35 & 0.4 \\ 0.5 & 0.55 \\ 0.65 & 0.7 \end{bmatrix} \quad r = \begin{bmatrix} 0.071 \\ 0.215 \\ 0.36 \\ 0.508 \\ 0.657 \end{bmatrix} \quad sol = \begin{bmatrix} -1.874 \cdot 10^{-13} \\ 0 \\ -2.331 \cdot 10^{-15} \\ 0 \\ 0 \end{bmatrix}$$

Define the expressions for the degrees of consolidation of layers I and II for case b.

$i := 12 \qquad INC := .1$

$$Ucon_1(\sigma,\mu,\nu,T) := \left| \begin{array}{l} sum \leftarrow 0 \\ W \leftarrow zbrak(i,\sigma,\mu,\nu,INC) \\ \text{for } n \in 1..i \\ \quad \left| \begin{array}{l} An \leftarrow brent\left(W_{n-1,0}, W_{n-1,1}, \sigma,\mu,\nu,TOL\right) \\ term \leftarrow \dfrac{\sin(An)\cdot\cos(An)\cdot\sin(\mu\cdot\nu\cdot An)}{An^2\cdot\left(\sigma\cdot\sin(\mu\cdot\nu\cdot An)^2 + \mu\cdot\nu\cdot\cos(An)^2\right)}\cdot\exp(-T\cdot An^2) \\ sum \leftarrow sum + term \end{array} \right. \\ 1 - 2\cdot sum \end{array} \right.$$

$$Ucon_2(\sigma,\mu,\nu,T) := \left| \begin{array}{l} sum \leftarrow 0 \\ W \leftarrow zbrak(i,\sigma,\mu,\nu,INC) \\ \text{for } n \in 1..i \\ \quad \left| \begin{array}{l} An \leftarrow brent\left(W_{n-1,0}, W_{n-1,1}, \sigma,\mu,\nu,TOL\right) \\ term \leftarrow \dfrac{\cos(An)^2\cdot(1-\cos(\mu\cdot\nu\cdot An))}{An^2\cdot\left(\sigma\cdot\sin(\mu\cdot\nu\cdot An)^2 + \mu\cdot\nu\cdot\cos(An)^2\right)}\cdot\exp(-T\cdot An^2) \\ sum \leftarrow sum + term \end{array} \right. \\ 1 - \dfrac{2}{\mu\cdot\nu}\cdot sum \end{array} \right.$$

Define expression for "resultant" degree of consolidation for both layers.

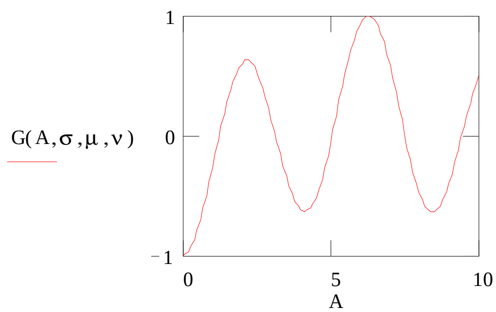
$$U_{res}(\sigma,\mu,\nu,T) := \left| \begin{array}{l} U1 \leftarrow Ucon_1(\sigma,\mu,\nu,T) \\ U2 \leftarrow Ucon_2(\sigma,\mu,\nu,T) \\ \dfrac{\nu\cdot U2 + U1}{1+\nu} \end{array} \right.$$

## Verification problem 2 for SAGE

$p := 100$  $\sigma := \frac{1}{2}$  $\mu := \frac{1}{2}$  $\nu := 1$  $h_1 := 5$  $e_1 = e_2$

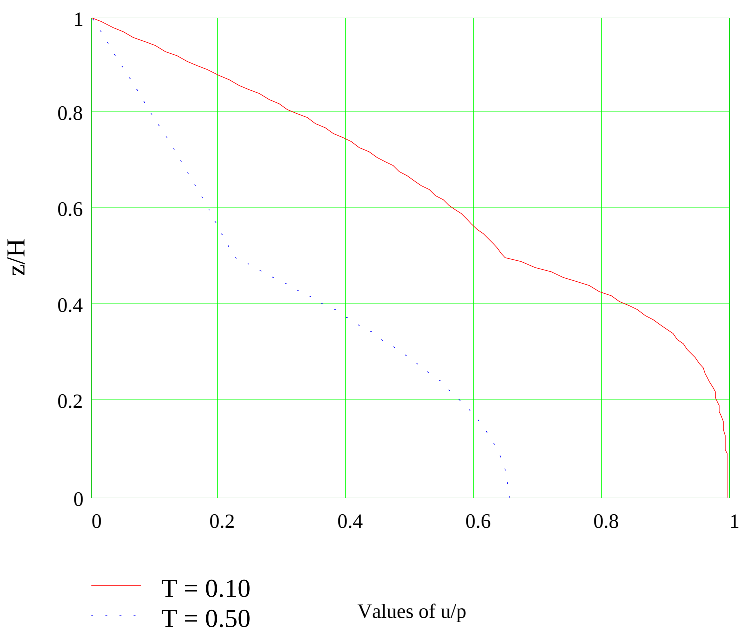
First, plot G(A) to find out nature of function for the parameters of the given problem.

$A := 0, .1 .. 10$




INC = 0.1 is a reasonable increment for the function **zbrak** to bracket the roots of G(A) for this problem. Note that INC must be small enough to "capture" each root of G(A).

$H := \text{Data}(p, \sigma, \mu, \nu, h_1, .1, 100)$  $I := \text{Data}(p, \sigma, \mu, \nu, h_1, .5, 100)$



```
T_trial(beg, end, n) :=  | T_val_0 ← 10^beg
                         | k ← 1
                         | for j ∈ beg .. end − 1
                         |    | num ← 10^j
                         |    | for jj ∈ 1 .. n
                         |    |    | T_val_k ← num · (10/n) · jj
                         |    |    | k ← k + 1
                         | T_val
```

Define the function $T_{trial}$ to generate the time factors, T, for a plot of resultant U versus the logarithm of T.

**beg** is the beginning log cycle (i.e. 0 for $10^0$)

**end** is the ending log cycle (i.e. 1 for $10^1$)

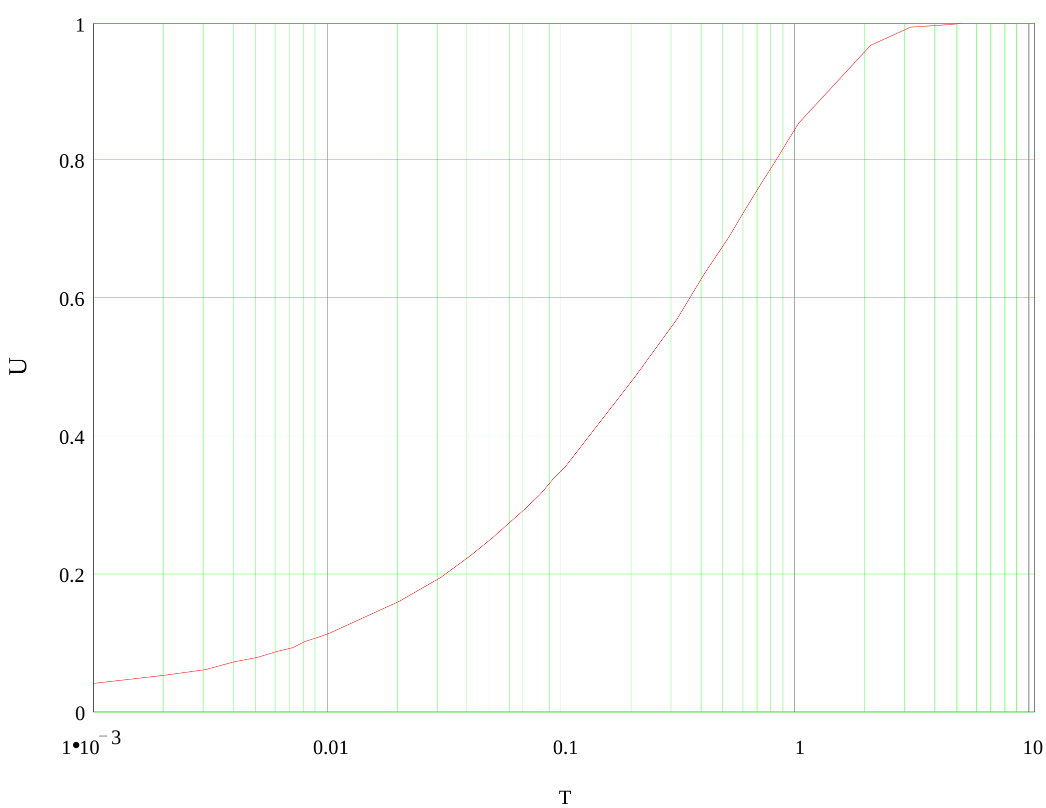
**n** is the number of increments for T in each log cycle

Example:

$$T_{trial}(0,2,3) = \begin{bmatrix} 1 \\ 3.333 \\ 6.667 \\ 10 \\ 33.333 \\ 66.667 \\ 100 \end{bmatrix}$$

$l := 0, 1 .. 40$ :range variable used for plot

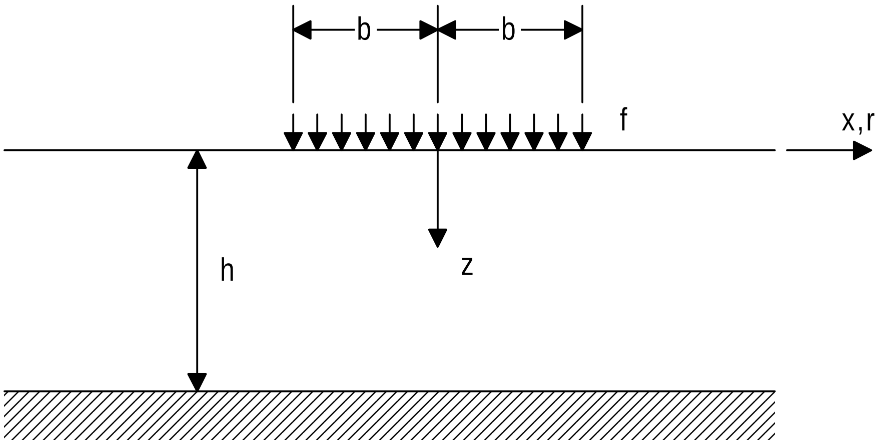
**Resultant U versus log T for Verification Problem 2**



## C.4. VERIFICATION PROBLEM THREE – SURFACE SETTLEMENT OF CLAY LAYER CONSOLIDATING UNDER A STRIP FOOTING LOAD

Gibson et al. (1970) derived an expression for the settlement of the surface of a clay layer loaded with a strip footing. The expression developed by Gibson et al. and its numerical evaluation are demonstrated in the following Mathcad® worksheet, Gibson et al.solution.mcd.

**The purpose of this MathCAD worksheet is to evaluate the expression for the surface settlement of a clay layer loaded with a strip footing. The original derivations were done by *Gibson, R.E., Schiffman, R.L.,and Pu, S.L. "Plane Strain and Axially Symmetric Consolidation of a Clay Layer on a Smooth Impervious Base," Quartly Journal of Mechanics and Applied Mathematics, Vol. XXIII, Pt. 4, 1970, pp. 505-520.* The expression that they developed is difficult to evaluate, because it involves an semi-infinite integral and an infinite series. Furthermore, it also involves finding the roots of a characteristic equation.**




Gibson et al found the following expression for the settlement of the surface of the clay layer.

$$w_o(x,t) = \frac{\eta \cdot f}{2 \cdot G} \cdot \int_0^{\infty} \Gamma(x,\lambda) \cdot \frac{\tanh(\lambda)^2}{\lambda^2} \cdot \left( \frac{2 \cdot \eta \cdot M(\lambda)}{2 \cdot \eta - 1} + P(\lambda,t) \right) d\lambda$$

<u>where:</u>

$$L(\lambda) = \left[ \lambda^2 \cdot \eta^{-1} \cdot (1 + \lambda \cdot \mathrm{csch}(\lambda) \cdot \mathrm{sech}(\lambda))^{-1} \right] - \lambda^2$$

$$M(\lambda) = \lambda \cdot \eta^{-1} \cdot \coth(\lambda) \cdot (1 + \lambda \cdot \mathrm{csch}(\lambda) \cdot \mathrm{sech}(\lambda))^{-1}$$

$$F(\alpha_n, \lambda) = \frac{1}{M(\lambda)} - \frac{1}{2} \cdot \left( 1 + \tan(\alpha_n)^2 + \alpha_n^{-1} \cdot \tan(\alpha_n) \right)$$

$$P(\lambda,t) = \sum_{n=1}^{\infty} \frac{\exp\left[ -\left( \alpha_n^2 + \lambda^2 \right) \cdot \frac{ct}{h^2} \right]}{F(\alpha_n, \lambda)}$$

$$\Gamma(x,\lambda) = \frac{2 \cdot h}{\pi \cdot \lambda} \cdot \sin\left( \lambda \cdot \frac{b}{h} \right) \cdot \cos\left( \lambda \cdot \frac{x}{h} \right)$$ plane strain

$$\Gamma(r,\lambda) = b \cdot J_1\left( \lambda \cdot \frac{b}{h} \right) \cdot J_0\left( \lambda \cdot \frac{r}{h} \right)$$ axisymmetric

$\alpha_n$ are the roots of the characteristic equation:

$$\alpha^2 - L(\lambda) = \alpha \cdot M(\lambda) \cdot \tan(\alpha)$$

$$\eta = \frac{1 - \nu}{1 - 2 \cdot \nu} \qquad G = \frac{E}{2 \cdot (1 + \nu)} \qquad c = 2 \cdot G \cdot \eta \cdot \frac{k}{\gamma_w}$$

<u>material parameters:</u>

$\nu$ = poisson's ratio

G = shear modulus

k = permeability of the clay

$\gamma_w$ = unit weight of pore fluid (water)

c = coefficient of consolidation

## For the purposes of an example problem:

$h := 4$ $\nu := 0.30$ $k := 0.0028$ $f := 1000$ $TOL := 1\cdot 10^{-12}$

$b := 4$ $E := 33333$ $\gamma_w := 62.4$ $\varepsilon_{mach} := 1\cdot 10^{-14}$

$\eta := \frac{1-\nu}{(1-2\cdot\nu)}$ $G := \frac{E}{2\cdot(1+\nu)}$ $c := 2\cdot G\cdot\eta\cdot\frac{k}{\gamma_w}$ $\varepsilon := 1\cdot 10^{-12}$

$$\Gamma(x,\lambda) := \frac{2\cdot h}{\pi\cdot\lambda}\cdot\sin\left(\lambda\cdot\frac{b}{h}\right)\cdot\cos\left(\lambda\cdot\frac{x}{h}\right)$$

**plane strain case**

$$L(\lambda) := \left[\frac{1}{\eta\cdot(1+\lambda\cdot\text{csch}(\lambda)\cdot\text{sech}(\lambda))} - 1\right]\cdot\lambda^2 \qquad M(\lambda) := \frac{\lambda\cdot\coth(\lambda)}{\eta\cdot(1+\lambda\cdot\text{csch}(\lambda)\cdot\text{sech}(\lambda))}$$

$$F(\alpha,\lambda) := \frac{1}{M(\lambda)} - \frac{1}{2}\cdot\left(1+\tan(\alpha)^2 + \frac{\tan(\alpha)}{\alpha}\right)$$

Determine the roots ,$\alpha_n$, of the characteristic equation $\alpha^2$ - L($\lambda$) - $\alpha$*M($\lambda$)*tan($\alpha$)

$$q(\alpha,\lambda) := \alpha^2 - L(\lambda) - \alpha\cdot M(\lambda)\cdot\tan(\alpha)$$

Define the function bisect which returns a vector containing a and b, where a and b define an interval (a,b) for which f(a)*f(b)<0, using the Bisection Method. For this case f(x) is q(a,l)

```
bisect(n, λ, ϕ, itmax) := | a ← (n − 1)·π
                          | b ← (n − 1/2)·π
                          | iter ← 0
                          | while iter < itmax
                          |    | xnew ← a + (b − a)/2
                          |    | iter ← iter + 1
                          |    | [a]
                          |    | [b]
                          |    | break if (b − a)/π < ϕ
                          |    | fa ← q(a, λ)
                          |    | fnew ← q(xnew, λ)
                          |    | (a ← xnew) if (fa·fnew > 0)
                          |    | (b ← xnew) if (fa·fnew ≤ 0)
                          | [a]
                          | [b]
```

The initial estimate of (a,b) is (n$\pi$,n+1/2$\pi$)

The function, bisect, will be used to determine a very small interval (a,b) which contains $\alpha_n$.

**Note**: $\alpha_1$ occurs in the interval (0,$\pi$/2), $\alpha_2$ occurs in the interval ($\pi$,3$\pi$/2) and so on.

Define the function **Muller** which uses Muller's method to refine the estimate of $\alpha_n$, the root of $q(\alpha,\lambda)$.

$$
\begin{aligned}
\text{Mulle}\left(r\ x_0, x_1, x_2, \text{maxit}, \lambda\right) := \ & \text{for } \text{iter} \in 1..\text{maxit} \\
& \quad h_0 \leftarrow x_1 - x_0 \\
& \quad h_1 \leftarrow x_2 - x_1 \\
& \quad P_0 \leftarrow q(x_0, \lambda) \\
& \quad P_1 \leftarrow q(x_1, \lambda) \\
& \quad P_2 \leftarrow q(x_2, \lambda) \\
& \quad \delta_0 \leftarrow \frac{P_1 - P_0}{h_0} \\
& \quad \delta_1 \leftarrow \frac{P_2 - P_1}{h_1} \\
& \quad a_0 \leftarrow P_2 \\
& \quad a_2 \leftarrow \frac{\delta_1 - \delta_0}{h_1 + h_0} \\
& \quad a_1 \leftarrow a_2 \cdot h_1 + \delta_1 \\
& \quad D \leftarrow \sqrt{a_1^{\,2} - 4 \cdot a_2 \cdot a_0} \\
& \quad E \leftarrow a_1 + D \ \text{ if } \left| a_1 + D \right| > \left| a_1 - D \right| \\
& \quad E \leftarrow a_1 - D \ \text{ otherwise} \\
& \quad h \leftarrow -2 \cdot \frac{a_0}{E} \\
& \quad \text{xstar} \leftarrow x_2 + h \\
& \quad x_0 \leftarrow x_1 \\
& \quad x_1 \leftarrow x_2 \\
& \quad x_2 \leftarrow \text{xstar} \\
& \text{xstar}
\end{aligned}
$$

Reference: Asaithambi, N.S. *Numerical Analysis Theory and Practice*. Saunders College Publishing.

Define the root finding function **alpha**. **alpha** finds the nth root of the characteristic equation: $q(\alpha,\lambda)$. **Alpha** uses the bisection method to narrow the interval around the nth root and then calls **Muller** to "polish" the root. Note: Muller's method finds real and complex roots.

$$\text{alpha}(n,\lambda) := \begin{array}{|l} x \leftarrow \text{bisect}(n,\lambda,\varepsilon,42) \\ \text{Re}\left(\text{Muller}\left(x_0, \frac{x_0 + x_1}{2}, x_1, 1, \lambda\right)\right) \end{array}$$

Example: $an := \text{alpha}(2,5)$ $\quad an = 4.314963 \quad q(an,5) = 1.776357 \cdot 10^{-14}$

Define the function, term, which evaluates the nth term of P(l,t).

$$\text{term}(n,\lambda,t) := \begin{array}{|l} \alpha_n \leftarrow \text{alpha}(n,\lambda) \\ \text{num} \leftarrow \exp\left[-\left[\left(\alpha_n\right)^2 + \lambda^2\right]\cdot\frac{c\cdot t}{h^2}\right] \\ \text{denom} \leftarrow F\left(\alpha_n,\lambda\right) \\ \frac{\text{num}}{\text{denom}} \end{array}$$

Define the function, $H(\lambda,t)$, which estimates $P(\lambda,t)$ by summing $P(\lambda,t)$ until the terms computed become insignificant.

$$H(\lambda,t) := \begin{array}{|l} H \leftarrow 0 \\ i \leftarrow 350 \\ s_i \leftarrow 0 \\ \text{check} \leftarrow 10\cdot TOL \\ \text{while } |\text{check}| > TOL \\ \quad \begin{array}{|l} i \leftarrow i - 1 \\ \left(s_i \leftarrow \text{term}(i,\lambda,t)\right) \\ \text{check} \leftarrow \frac{s_i}{H} \text{ if } |H| > 0 \\ H \leftarrow H + s_i \\ \text{break if } i = 1 \end{array} \\ H \end{array}$$

Note that roundoff and truncation errors pose serious problems to the evaluation of $P(\lambda,t)$ using $H(\lambda,t)$. Next, Euler's transformation of alternating series and van Wijngaarden's implementation will be examined as an alternative way of approximating $P(\lambda,t)$.

Define the function Euler, which evaluates Euler's transformation of alternating series with van Wijngaarden's implementation. One term of the original alternating series are incorporated into the estimates of the partial differences.

```
sub1(nterm, wk1, wk2, sum) := | if |wk2| ≤ |wk1|
                              |    | newsum ← sum + 0.5·wk2
                              |    | n ← nterm + 1
                              | otherwise
                              |    | newsum ← sum + wk2
                              |    | n ← nterm
                              | ( n      )
                              | ( newsum )
```

**Sub1** is a subroutine used by Euler. **Sub1** determines whether a new partial difference should be evaluated (increase nterm by 1) or just revise the estimate of S.

```
Euler(term, jterm, wksp) := | if jterm = 1
                            |    | n ← 1
                            |    | sum ← 0.5·term
                            |    | wksp_2 ← term
                            | otherwise
                            |    | n ← wksp_0
                            |    | tmp ← wksp_2
                            |    | wksp_2 ← term
                            |    | for j ∈ 1..n − 1   if n ≥ 2
                            |    |    | dum ← wksp_(j+2)
                            |    |    | wksp_(j+2) ← 0.5·(wksp_(j+1) + tmp)
                            |    |    | tmp ← dum
                            |    | wksp_(n+2) ← 0.5·(wksp_(n+1) + tmp)
                            |    | a ← sub1(n, wksp_(n+1), wksp_(n+2), wksp_1)
                            |    | n ← a_0
                            |    | sum ← a_1
                            | wksp_0 ← n
                            | wksp_1 ← sum
                            | wksp
```

**Euler** estimates the summation of a convergent infinite series whose terms alternate in sign.

Wijngaarden's implementation adapts Euler's transformation to positive or negative convergent series.

Reference: Press, W.H., Teukolosky, S. A., Vetterling, W.T., and Flannery, B.P. *Numerical Recipes in FORTRAN.* 2nd ed. Cambridge University Press.

Set up a trial series to evaluate the function Euler. Solve the problem: $A = \sum_{n=1}^{\infty} (k)^n$

for k<1: $A = \frac{k}{1-k}$

```
trial(k) := | j ← 0
            | s ← 0
            | W ← 0
            | check ← 10·TOL
            | W_old ← 0
            | while |check| > TOL
            |   | j ← j + 1
            |   | s ← 0
            |   | for t ∈ 8, 7 .. 0
            |   |   s ← s + 2^t·k^(2^t·j)
            |   | s ← (-1)^(j-1)·s
            |   | W ← Euler(s, j, W)
            |   | check ← (W_1 - W_old)/W_old  if |W_old| > 0
            |   | W_old ← W_1
            | [W_1]
            | [j  ]
```

The function **trial** uses the Euler function to estimate A.

Use van Wijngaarden's procedure for evaluating a positive series with Euler's technique (i.e. convert the series into an alternating series)

```
Asum(k, n) := | s ← 0
              | for j ∈ 1 .. n
              |   s ← s + k^j
```

**Asum** evaluates A through n terms.

$\text{Asum}\left(\frac{1}{4}, 22\right) = 0.333333333333314$ $\quad \text{trial}\left(\frac{1}{4}\right) = \begin{bmatrix} 0.333333333333335 \\ 22 \end{bmatrix}$ $\quad A = \frac{1}{3}$ for k = 1/4

This verifies that **Euler** works, but it does not offer significant savings in computations in this particular case.

Define J(λ,t), which will estimate P(λ,t) using the Euler's transformation. Van Wijngaarden's transformation is used to convert P(λ,t) into an alternating series (terms in the sum alternate in sign). Euler's transformation only works for alternating series.

```
J(λ, t) :=  W ← 0
            check ← 10·TOL
            jterm ← 0
            Σ ← 0
            r ← 0
            while |check| > TOL
                r ← r + 1
                jterm ← jterm + 1
                w_r ← 0
                for k ∈ 12, 11 .. 0
                    w_r ← w_r + 2^k·term(2^k·r, λ, t)
                w_r ← (−1)^(jterm − 1)·w_r
                W ← Euler(w_r, jterm, W)
                check ← (W_1 − Σ)/|Σ|  if |Σ| > 0
                Σ ← W_1
            [ Σ ]
            [ r ]
```

Examples of estimating P(λ,t) with J(λ,t) and H(λ,t):

$J(10, 0) = \begin{bmatrix} -2.284385 \\ 23 \end{bmatrix}$ $H(10, 0) = -2.266755$ $J(.5, 0) = \begin{bmatrix} -0.13361 \\ 24 \end{bmatrix}$ $H(.5, 0) = -0.133549$

$J(10, 1) = \begin{bmatrix} -2.36028 \cdot 10^{-7} \\ 7 \end{bmatrix}$ $H(10, 1) = -2.36028 \cdot 10^{-7}$

$J(.5, 1) = \begin{bmatrix} -0.087142 \\ 6 \end{bmatrix}$ $H(.5, 1) = -0.087142$

It is apparent that the functions J and H give slightly different results at times early in the solution, but there is almost no difference in the values later in the solution. The source of this difference is probably round-off error. The round-off is probably larger for H since it is a simple summation.

Define the function Gauleg which will return a matrix whose 1st column contains the Gauss points and whose 2nd column contains the Gauss weights for the m-point quadrature rule for the interval (a,b)

```
Gauleg(a, b, m) := | j ← (m + 1)/2
                   | eps ← 3·10^-14
                   | xm ← 0.5·(a + b)
                   | xl ← 0.5·(b − a)
                   | for i ∈ 1 .. j
                   |   | z ← cos(π·(i − .25)/(m + .5))
                   |   | check ← 2·eps
                   |   | while check > eps
                   |   |   | p1 ← 1
                   |   |   | p2 ← 0
                   |   |   | for k ∈ 1 .. m
                   |   |   |   | p3 ← p2
                   |   |   |   | p2 ← p1
                   |   |   |   | p1 ← ((2·k − 1)·z·p2 − (k − 1)·p3)/k
                   |   |   | pp ← m·(z·p1 − p2)/(z·z − 1)
                   |   |   | z1 ← z
                   |   |   | z ← z1 − p1/pp
                   |   |   | check ← |z − z1|
                   |   | A[i−1,0] ← xm − xl·z
                   |   | A[m−i,0] ← xm + xl·z
                   |   | A[i−1,1] ← 2·xl/((1 − z·z)·pp·pp)
                   |   | A[m−i,1] ← A[i−1,1]
                   | A
```

Examples:

$$\text{Gauleg}(-1,1,2) = \begin{bmatrix} -0.5773502692 & 1 \\ 0.5773502692 & 1 \end{bmatrix} \qquad \text{Gauleg}(0,\pi,3) = \begin{bmatrix} 0.354062724 & 0.872664626 \\ 1.5707963268 & 1.3962634016 \\ 2.7875299296 & 0.872664626 \end{bmatrix}$$

**Now, define a MathCAD function, w, to evaluate the settlement, w, at any (x,t). The variable quad control how the integration is performed. Gauss-Legendre quadrature with a quad-point rule is used. The integral is broken up into subintegrals each of length π. This is done because the period of the function, w, is π.**

$$
\begin{array}{l}
w(x,t,quad) := \\
\quad w \leftarrow 0 \\
\quad i \leftarrow 0 \\
\quad GAUS \leftarrow Gauleg(0,\pi,quad) \\
\quad check \leftarrow 1 \\
\quad \text{while } |check| > 1\cdot 10^{-3} \\
\qquad i \leftarrow i+1 \\
\qquad a \leftarrow (i-1)\cdot\pi \\
\qquad b \leftarrow i\cdot\pi \\
\qquad sum \leftarrow 0 \\
\qquad \text{for } j \in 1..quad \\
\qquad\quad weight \leftarrow GAUS_{j-1,1} \\
\qquad\quad \lambda \leftarrow GAUS_{j-1,0} + a \\
\qquad\quad sum \leftarrow sum + \Gamma(x,\lambda)\cdot\dfrac{\tanh(\lambda)^2}{\lambda^2}\cdot\left(\dfrac{2\cdot\eta\cdot M(\lambda)}{2\cdot\eta - 1} + J(\lambda,t)_0\right)\cdot weight \\
\qquad w \leftarrow w + sum \\
\qquad check \leftarrow \dfrac{sum}{w} \\
\quad w \leftarrow \dfrac{\eta\cdot f}{2\cdot G}\cdot w
\end{array}
$$

$$w(0,0,5) = 0.079367$$

Define two functions:

$w_{final}$ - evaluates the final settlement at x

$w_{immed}$ - evaluates the immediate settlement at x

```
w_final(x, n, quad) :=  w ← 0
                        check ← 1
                        for i ∈ 1 .. 2·n
                            a ← (i − 1)·π/2
                            b ← i·π/2
                            GAUS ← Gauleg(a, b, quad)
                            sum ← 0
                            for j ∈ 1 .. quad
                                weight ← GAUS[j−1,1]
                                λ ← GAUS[j−1,0]
                                sum ← sum + Γ(x,λ)·tanh(λ) / (λ·(1 + λ·csch(λ)·sech(λ)))·weight
                            w ← w + sum
                            check ← sum/w
                        w ← η·f/((2·η − 1)·G)·w
```

$$w_{immed}(x,n,quad) := w_{final}(x,n,quad)\cdot\frac{2\cdot\eta-1}{2\cdot\eta}$$

For our example problem:

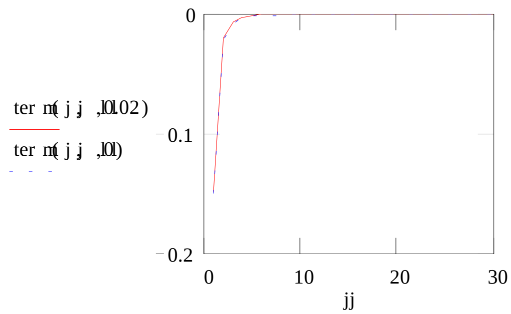
$$w_{final}(0,20,5) = 0.111079 \qquad \frac{G\cdot w_{final}(0,20,5)}{b\cdot f} = 0.356018$$

$$w_{immed}(0,5,5) = 0.07976 \qquad \frac{G\cdot w_{immed}(0,20,5)}{b\cdot f} = 0.254298$$

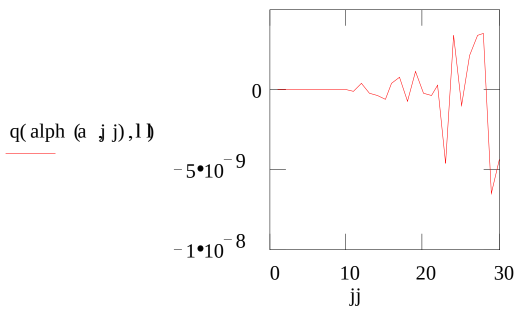
These results agree with Figure 5 from Gibson et al, 1970.

The following pages contain plots that investigate the nature of the functions involved in Gibson et al's solution.

$jj := 1 .. 30$ $\quad ll := 1$

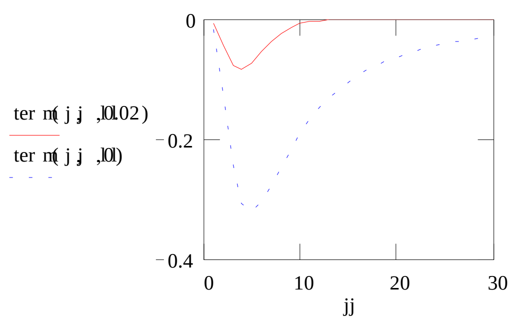



Plot of the first 30 terms of the infinite series P(λ,t). for λ = 1and t = 0 and 0.02




The effects of noise and roundoff error appear in the higher values of $\alpha_n$ (~n>10). λ = 1

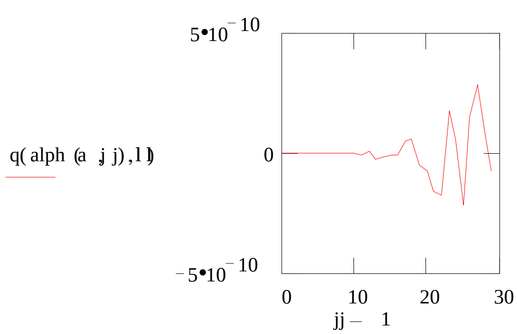
$ll := 20$ Now set λ = 20 and replot the data.

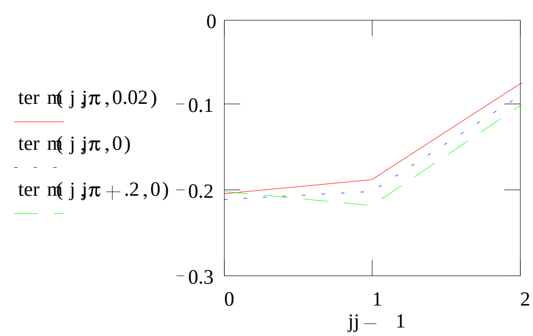



$J(ll, 0) = \begin{bmatrix} -4.566113 \\ 25 \end{bmatrix}$ $\quad J(ll, .02) = \begin{bmatrix} -0.428654 \\ 29 \end{bmatrix}$

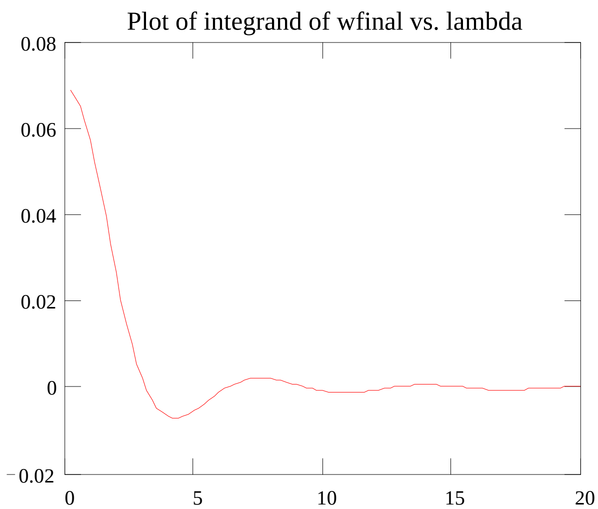
$H(ll, 0) = -4.4956$ $\quad H(ll, .02) = -0.428654$

Note the "pulse" type shape of the plot of the infinite series P(λ,t). This "pulse" shape appears when λ is greater than π.



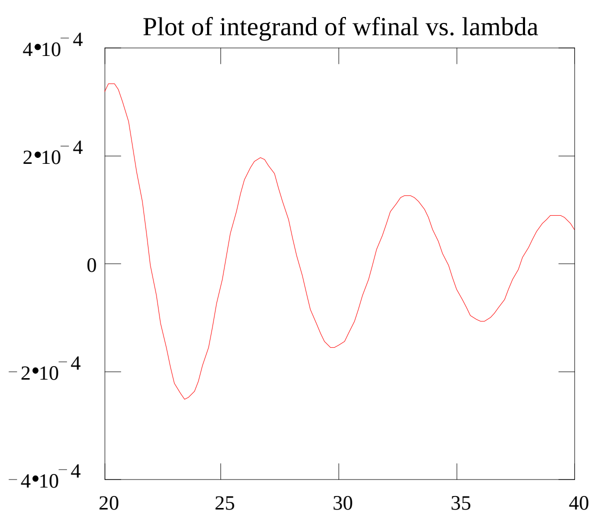



ll := 0, .2 .. 20




This plot clearly shows the damped oscillating nature of the integrand of Gibson et al's expression for settlement.

ll := 20, 20.2 .. 40




SAGE results are compared with this solution in Chapter 3.