THE DETERMINATION OF THE LAW OF VARIATION FOR THE TEMPERATURE OF THE COKING CHAMBER`S WALL

RIZEA Nicoleta, FLOREA Ion
Petroleum-Gas University of Ploiesti
nicoletarizea @yahoo.com

Keywords: The metallic wall of coking chamber, the thermic-oligo-cyclic fatigue, theT(P,h) function, the third order Spline Functions, experimental measurements

Abstract: The most important input data for the Finite Element Method programs, in this case of a coking chamber, is the value of the temperature in the metallic wall, T, which is distributed depending on the chamber's height h and the time percent of the coking cycle P. So, it was conceived a calculus program using the interpolation method with the third order Spline Functions, which established the numerical expression T(P,h) function, and it is recognized by the program in it's mathematical form.

1. INTRODUCTION

The coking chamber from the coking battery works under many complex loads, which have a periodical variation. These loads can have a mechanical, thermal or dynamic character. Concerning the thermal loads, the cyclic thermal mode of the coking chamber has the temperature's maintaining bearing bigger than the creep temperature.
The thermocyclic working of the coking chambers complicates the description of the metallic wall's thermal field. Industrial experimental research, made on the working coking chambers, gave the possibility to realize useful temperature maps, in order to have a better control of the temperature during a coking cycle. All these maps were realized based on the statistical processing of the obtained data.

2. THE CHAMBER`S WALL TEMPERATURE VARIATION ALONG THE HEIGHT AND DURING A COKING CYCLE

In figure 1, the chamber's wall temperature variation along the height and during a cycle is presented. Depending on its height, the coking chamber is divided in three sections- I, II, III (see figure 1.b), each of them being divided, also, in subsections from 0 to 8. Depending on the relative duration of the coking cycle, the lengths of time, specific for a coking cycle (I- IV), are determined [1],[3].
Another problem to be studied is the heating up and cooling off speeds of the metallic wall, i.e. the gradient of the temperature variation's speed as a function of the coking cycle's relative duration and of the coking chamber's height.
The coking chamber stress and strain analysis involves the 3D simulation, with a maximum dimensional and dividing precision, due to the fact that the coking chamber is submitted to mechanical, thermocyclic and wind loads.
The coking chamber was prepared for the finite element analysis by dividing its complex geometric shapes in solid-finite elements (hexahedron cells). The choice of the finite element was made taking into account the material's thickness of the complex shells and, also, the fact that, per total, a volume of over 48 000 cubic metres is divided. The tri-dimensional simulation could be realized using advanced 3D computer programs, which allow the data transfer to advanced simulation programs [2], [8], [9].
Considering the large volume of work (five million finite elements and forty million nodes), the approximations could vitiate the results, which are obtained after a long duration simulation process. We must specify that the phenomena of cycling stress imposed the use of the software simulation module Cosmos M 2007, for the fatigue phenomenon.

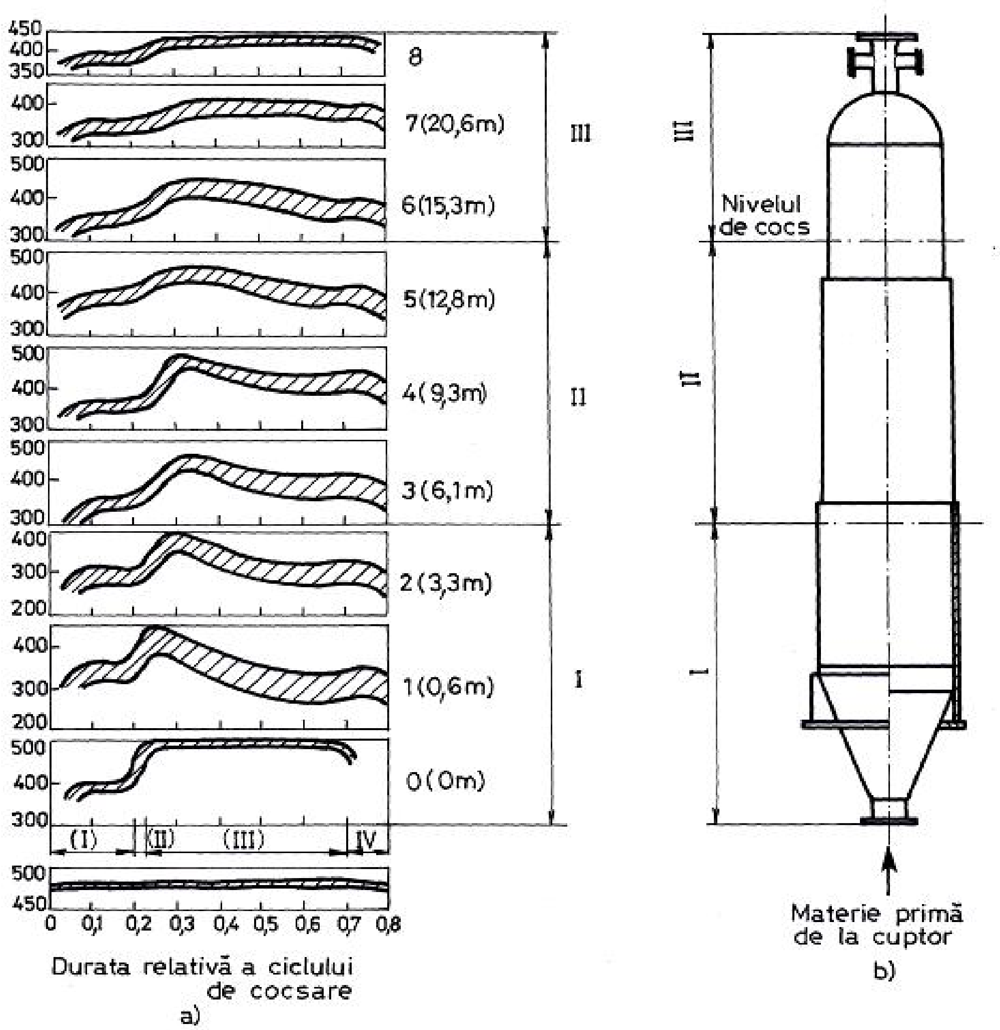


***Figure 1**: The chamber's wall temperature variation along the height and during a cycle*

So, the most important input data for the finite element programs, in the case of the coking chamber, is the value of the metallic wall's temperature, *T*, which is distributed according to the coking chamber's height, *h*, and to the percentage of time from the coking cycle.
Being one of the essential input data in creating the finite element simulation, this law of variation is determined from experimental measurements, realized during the plant's working, by using temperature sensors (thermocouples). These sensors are placed against the metallic wall of the coking chamber (figure 2), protected by thermal insulation, and this thing is stipulated in the design phase (according to the production drawings and attachment drawings). These sensors have a control role of the process evolution and give the possibility of the real-time temperature measurement.
At the coking chamber`s level, on sections I and II, where the raw material is transformed into coke, seven sensors (Tr 1,…, Tr 7) are placed. We specify that the first sensor Tr 1 is placed at the joint R3 level, considered to be at height zero, and the sensor Tr 7 is placed at the level corresponding to the free surface of the liquid, considered here to be at the joint R8f level (see fig. 2). There are another two sensors placed at the superior part of the coking chamber, one against the exterior surface of the spherical cover, Tr 8, and the

other against the dome, Tr 9. The vertical mounting dimensions for the sensors` placement are presented in Table 1.

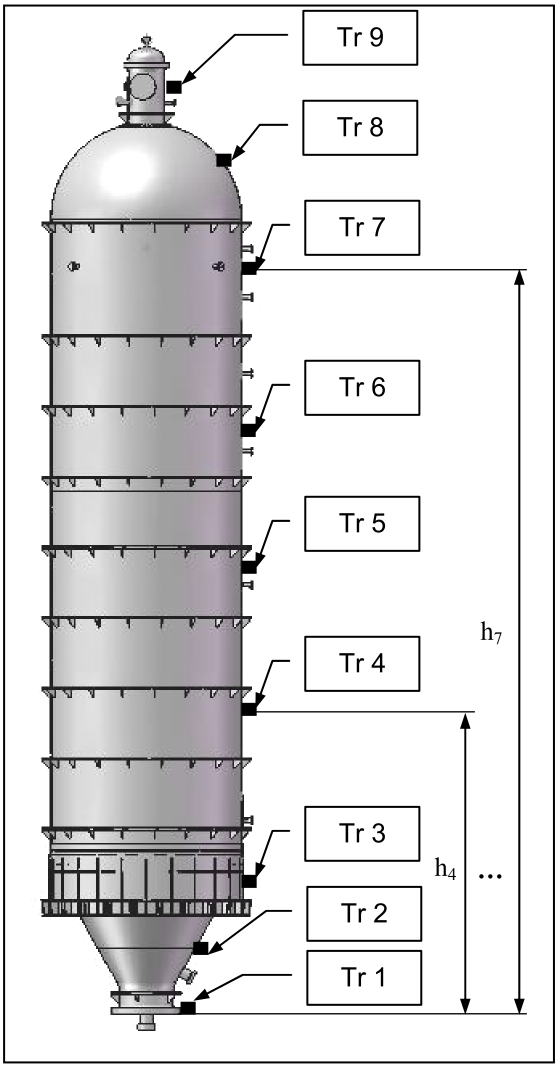


Figure 2: The vertical mounting dimensions for the sensors` placement

Table 1

Tr_k	h_k [mm]	Tr_k	h_k [mm]
Tr 1	0	Tr 6	12800
Tr 2	700	Tr 7	21650
Tr 3	3500	Tr 8	24500
Tr 4	7600	Tr 9	27500
Tr 5	9500		

The use of these sensors allows the realization of a database, which can make a connection between the temperatures *T*, the relative duration of the coking cycle, expressed here as percentage *P* and the interior height of the coking chamber *h*. After analyzing the recorded data, one obtained the mean values of the temperature, function of the percentage *P* from the coking cycle and of the height dimension *h*. The results of the obtained experimental measurements, processed using Excel 2007 and completed by addition, are presented in [1].

3. PRESENTATION OF THE PROGRAM FOR DETERMINING THE FUNCTION T(P,h)

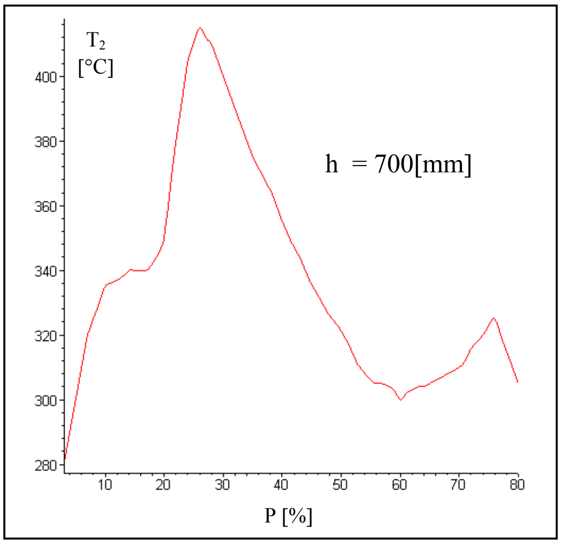
After that, the question is the finding of a continuous analytical function, which can establish the connection between these ones (the temperature *T*, the relative duration of the coking cycle *P* and the dimension *h*). Practically, the analytical formula of the function *T*(*P*, *h*) can be found only in a numerical form and, for this reason, it is proposed below its determination by a double numerical interpolation, using the interpolation method with Spline functions of order 3.

In this sense, having as input the results of the experimental measurements from [1]., it was developed the numerical program, written in the programming language of the mathematical software Maple 11, which sets this function. The experimental measurements are also presented into the next program:

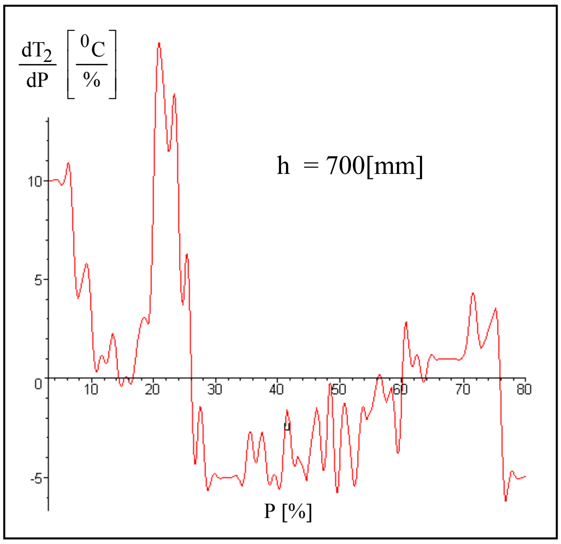
```
➢ restart:
➢ with(CurveFitting):with(plots):
➢ P_dat:=60: H_dat:=2.75: H_min:=9: H_max:=10: n:=20:
➢ P:=[seq(i,i=3..80)]:
➢ L_h:=[0,0.6,3.3,6.1,9.3,12.8,15.3,20.6,27.5]:
➢ LT_h1:=[330,345,360,370,380,385,390,392,393,394,395,396,397,398,399,400,405,420,435,455,475
  ,480,485,490,495,495,495,495,495,495,495,495,495,495,495,495,495,495,495,495,495,495,495,49
  5,495,495,495,495,495,495,495,495,495,495,495,495,495,495,495,495,495,495,492,491,490,488,4
  86,480,475,465,450,440,430,420,415,410,405,400]:nops(LT_h1):
➢ LT_h2:=[280,290,300,310,320,325,330,335,336,337,338,340,340,340,340,342,345,350,365,380,392
  ,405,410,415,412,410,405,400,395,390,385,380,375,372,368,365,360,355,350,348,344,340,335,33
  2,330,326,325,320,318,315,310,308,306,305,305,304,303,300,302,303,304,304,305,306,307,308,3
  09,310,312,316,318,320,323,325,320,315,310,305]:nops(LT_h2):
➢ LT_h3:=[230,240,250,260,270,280,285,286,288,290,290,290,290,290,290,290,290,290,294,296,300
  ,310,325,335,350,365,375,380,378,375,370,368,365,360,355,350,348,340,335,330,325,320,317,31
  5,312,310,305,303,302,300,300,298,296,295,294,293,292,292,291,290,291,294,295,298,300,300,3
  00,302,304,305,303,298,296,294,290,284,275,270]:nops(LT_h3):
➢ LT_h4:=[290,292,295,300,312,324,330,335,340,342,343,344,345,346,347,348,350,355,358,362,368
  ,373,380,390,403,410,420,430,440,442,444,448,446,444,438,436,430,428,426,420,415,410,406,40
  4,400,398,397,396,394,390,390,388,387,386,385,384,382,380,380,380,380,380,380,382,384,386,3
  87,388,386,382,381,380,378,376,375,370,366,362]:nops(LT_h4):
➢ LT_h5:=[310,320,330,336,342,348,350,352,354,356,358,360,358,356,358,360,361,362,364,367,370
  ,375,385,400,410,430,445,460,464,470,468,466,462,460,458,454,450,448,446,442,441,440,436,43
  4,430,425,423,420,418,416,414,412,410,407,408,409,410,410,410,410,410,410,410,411,412,413,4
  14,415,416,418,420,422,420,418,414,410,404,398]:nops(LT_h5):
➢ LT_h6:=[350,360,365,370,375,380,382,386,388,390,392,394,395,396,397,397,398,399,400,404,410
  ,420,424,430,432,434,436,438,440,441,442,443,444,446,445,442,440,438,437,436,435,433,430,42
  8,426,424,420,418,414,410,408,406,404,400,398,396,397,395,394,393,392,391,390,390,390,390,3
  90,390,392,397,394,392,390,388,384,380,375,370]:nops(LT_h6):
➢ LT_h7:=[310,315,325,335,340,343,345,348,350,352,353,354,356,357,358,360,362,365,370,372,378
  ,380,384,390,400,410,415,420,422,426,428,426,425,424,422,421,420,419,418,417,416,415,414,41
  2,410,408,406,402,400,398,397,396,395,394,392,390,388,386,382,380,378,377,376,375,375,375,3
  75,375,375,375,375,375,374,372,371,370,366,360]:nops(LT_h7):
➢ LT_h8:=[310,315,320,330,336,338,340,343,348,348,348,348,348,348,348,348,348,350,352,354,355
  ,356,357,358,360,365,370,374,376,380,382,385,386,387,390,390,390,390,390,390,390,390,390,39
  0,390,390,390,390,390,390,390,390,390,390,390,390,390,390,388,386,385,382,381,380,378,376,3
  75,375,375,375,375,375,375,374,372,370,365,360]:nops(LT_h8):
➢ LT_h9:=[360,365,370,374,377,378,379,380,380,380,380,380,380,381,382,383,384,386,390,392,396
  ,400,406,414,418,420,420,420,420,420,420,420,420,420,420,420,420,420,420,420,420,420,420,42
  0,420,420,420,420,422,423,425,426,428,430,430,430,430,430,430,430,430,430,430,430,430,430,4
  30,430,428,425,422,420,417,414,410,405,400,390]:nops(LT_h9):
➢ T_h1:=u-> Spline([seq([P[k],LT_h1[k]],k=1..nops(P))], u,degree=3): plot(T_h1(x),x=P[1]..P[nops(P)]):
➢ T_h2:=u-> Spline([seq([P[k],LT_h2[k]],k=1..nops(P))], u,degree=3): plot(T_h2(x),x=P[1]..P[nops(P)]):
➢ T_h3:=u-> Spline([seq([P[k],LT_h3[k]],k=1..nops(P))], u,degree=3): plot(T_h3(x),x=P[1]..P[nops(P)]):
➢ T_h4:=u-> Spline([seq([P[k],LT_h4[k]],k=1..nops(P))], u,degree=3): plot(T_h4(x),x=P[1]..P[nops(P)]):
➢ T_h5:=u-> Spline([seq([P[k],LT_h5[k]],k=1..nops(P))], u,degree=3): plot(T_h5(x),x=P[1]..P[nops(P)]):
➢ T_h6:=u-> Spline([seq([P[k],LT_h6[k]],k=1..nops(P))], u,degree=3): plot(T_h6(x),x=P[1]..P[nops(P)]):
```

- T_h7:=u-> Spline([seq([P[k],LT_h7[k]],k=1..nops(P))], u,degree=3): plot(T_h7(x),x=P[1]..P[nops(P)]):
- T_h8:=u-> Spline([seq([P[k],LT_h8[k]],k=1..nops(P))], u,degree=3): plot(T_h8(x),x=P[1]..P[nops(P)]):
- T_h9:=u-> Spline([seq([P[k],LT_h9[k]],k=1..nops(P))], u,degree=3): plot(T_h9(x),x=P[1]..P[nops(P)]):
- plot({T_h1(x),T_h2(x),T_h3(x),T_h4(x),T_h5(x),T_h6(x),T_h7(x),T_h8(x),T_h9(x)},x=P[1]..P[nops(P)]);
- LT_h:=[T_h1(P_dat),T_h2(P_dat),T_h3(P_dat),T_h4(P_dat),T_h5(P_dat),T_h6(P_dat),T_h7(P_dat), T_h8(P_dat),T_h9(P_dat)]:
- T:=evalf(Spline([seq([L_h[k],LT_h[k]],k=1..nops(L_h))],h, degree=3)):
- plot(T, h=L_h[1]..L_h[nops(L_h)]);
- dT_h1:=evalf(diff(Spline([seq([P[k],LT_h1[k]],k=1..nops(P))], u,degree=3),u)):plot(dT_h1,u=P[1]..P[nops(P)]):
- dT_h2:=evalf(diff(Spline([seq([P[k],LT_h2[k]],k=1..nops(P))], u,degree=3),u)):plot(dT_h2,u=P[1]..P[nops(P)]):
- dT_h3:=evalf(diff(Spline([seq([P[k],LT_h3[k]],k=1..nops(P))], u,degree=3),u)):plot(dT_h3,u=P[1]..P[nops(P)]):
- dT_h4:=evalf(diff(Spline([seq([P[k],LT_h4[k]],k=1..nops(P))], u,degree=3),u)):plot(dT_h4,u=P[1]..P[nops(P)]):
- dT_h5:=evalf(diff(Spline([seq([P[k],LT_h5[k]],k=1..nops(P))], u,degree=3),u)):plot(dT_h5,u=P[1]..P[nops(P)]):
- dT_h6:=evalf(diff(Spline([seq([P[k],LT_h6[k]],k=1..nops(P))], u,degree=3),u)):plot(dT_h6,u=P[1]..P[nops(P)]):
- dT_h7:=evalf(diff(Spline([seq([P[k],LT_h7[k]],k=1..nops(P))], u,degree=3),u)):plot(dT_h7,u=P[1]..P[nops(P)]):
- dT_h8:=evalf(diff(Spline([seq([P[k],LT_h8[k]],k=1..nops(P))], u,degree=3),u)):plot(dT_h8,u=P[1]..P[nops(P)]):
- dT_h9:=evalf(diff(Spline([seq([P[k],LT_h9[k]],k=1..nops(P))], u,degree=3),u)):plot(dT_h9,u=P[1]..P[nops(P)]):
- plot({dT_h1,dT_h2,dT_h3,dT_h4,dT_h5,dT_h6,dT_h7,dT_h8,dT_h9},u=P[1]..P[nops(P)]):
- dT:=evalf(diff(Spline([seq([L_h[k],LT_h[k]],k=1..nops(L_h))],h, degree=3),h)):
- plot(dT, h=L_h[1]..L_h[nops(L_h)]);
- H_total:=H_max-H_min:
- Delta_H:=evalf(H_total/n):
- L_intervale_H:=evalf([seq([H_min+(k-1)*Delta_H,H_min+k*Delta_H], k=1..n)]):
- L_Tmed_intervale:=[evalf(seq(int(T,h=H_min+(k-1)*Delta_H..H_min+k*Delta_H)/Delta_H,k=1..n))]:
- LIH:=[seq(L_intervale_H[k][1]..L_intervale_H[k][2],k=1..n)]:
- LTmin_intervale:=[seq(minimize(T, h=LIH[k]),k=1..nops(LIH))]:
- LTmax_intervale:=[seq(maximize(T, h=LIH[k]),k=1..nops(LIH))]:
- 'P_dat'=P_dat;'H_min'=H_min,'H_max'=H_max,'n_intervale'=n,'Delta_H'=Delta_H;seq(['Hmin'[k]=L_intervale_H[k][1],'Hmax'[k]=L_intervale_H[k][2],'Tmin'[k]=LTmin_intervale[k],'Tmax'[k]=LTmax_intervale[k],'T_med'[k]=L_Tmed_intervale[k]],k=1..n);
- FT_punctuala:=h-> evalf(Spline([seq([L_h[k],LT_h[k]],k=1..nops(L_h))],h, degree=3)):
- T_punctuala:=evalf(FT_punctuala(H_dat));

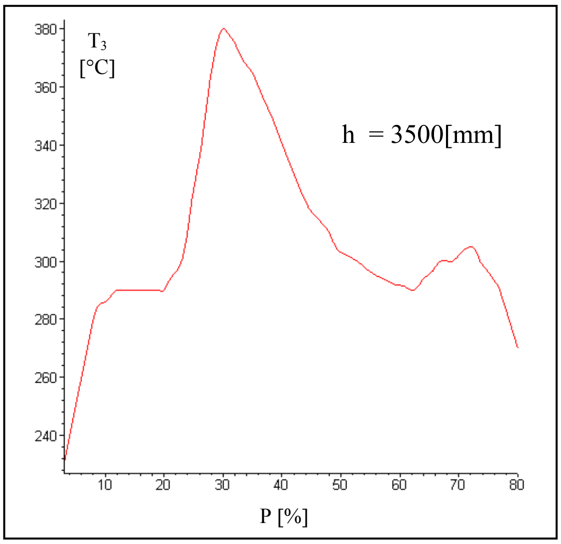
The program gives the analytical formulae of the functions *T* (*P*, *h*=cons.) and heir graphs, figures 3.a) - h). The program gives also the functions *T* (*P*=const., *h*).
It is of interest to study, also, the gradient of the temperature's variation speed function of the relative duration of the coking cycle, *P*, and function of the coking chamber's height, $\frac{d\left[T_k(P)\big|_{h_K=\text{cons}\tan t}\right]}{dP}$. In this situation, the program makes a numerical calculation and traces the graphs of the determined analytical functions (see figures 4.a) - h)).
In figures 3 and 4 this graphs are plotted separately.



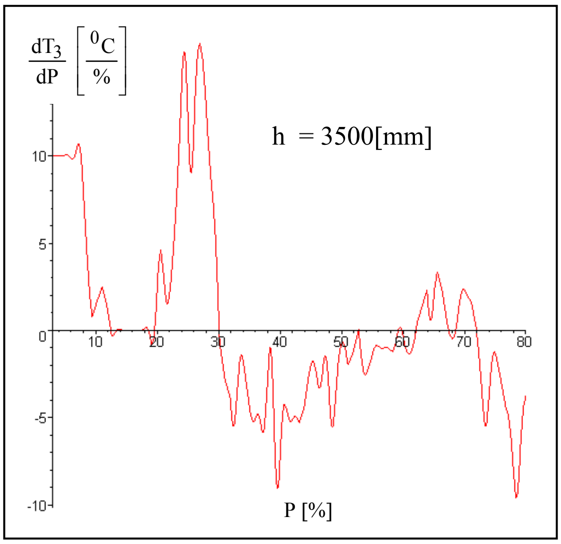
3.a)



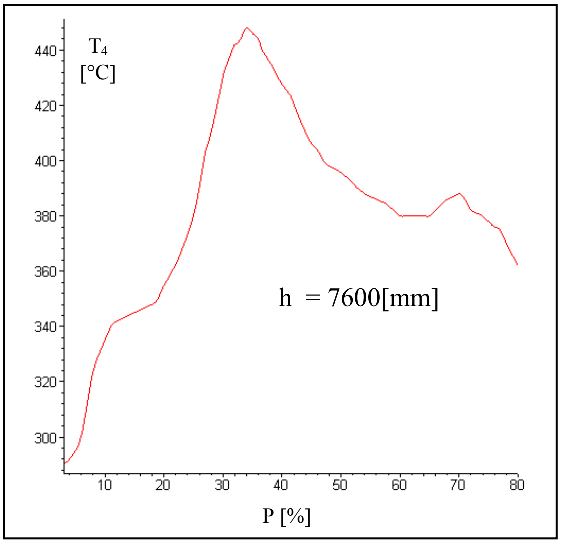
4.a)



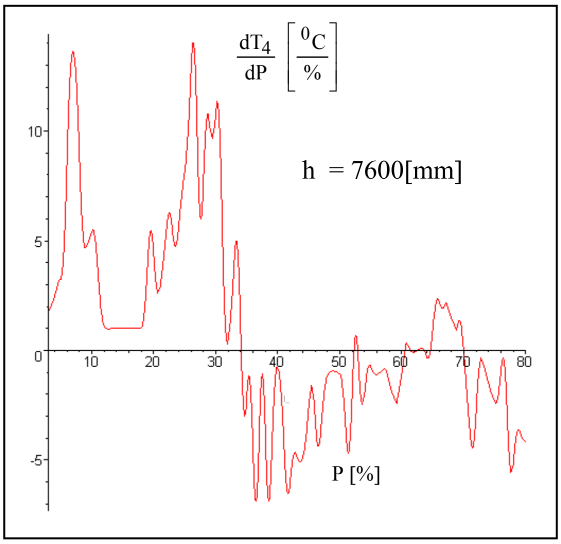
3.b)



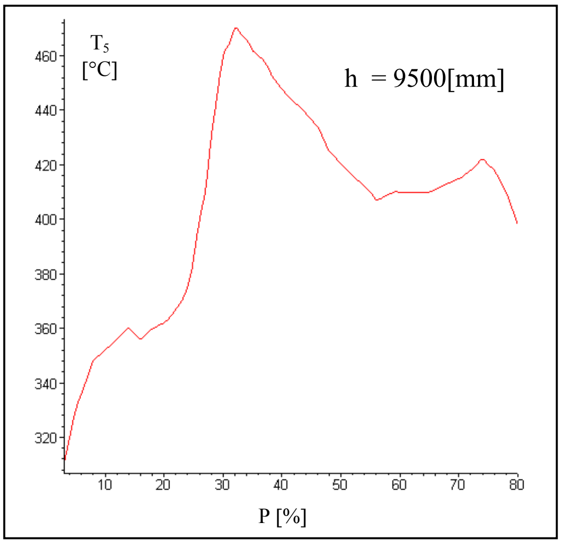
4.b)



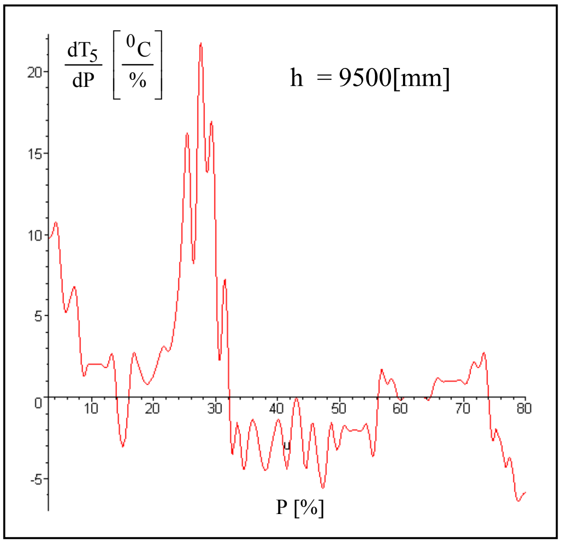
3.c)



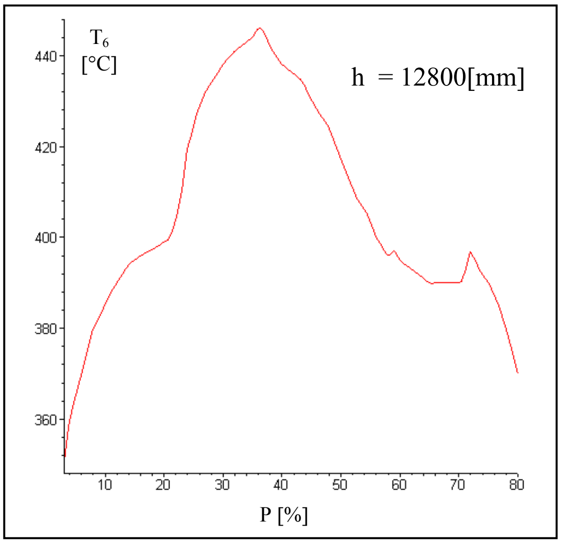
4.c)



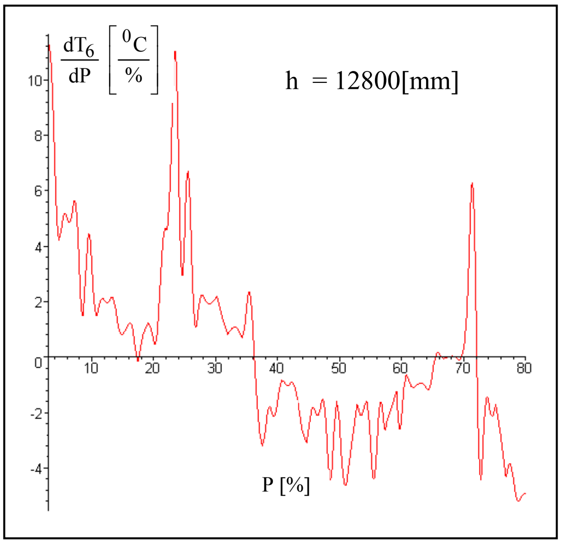
3.d)



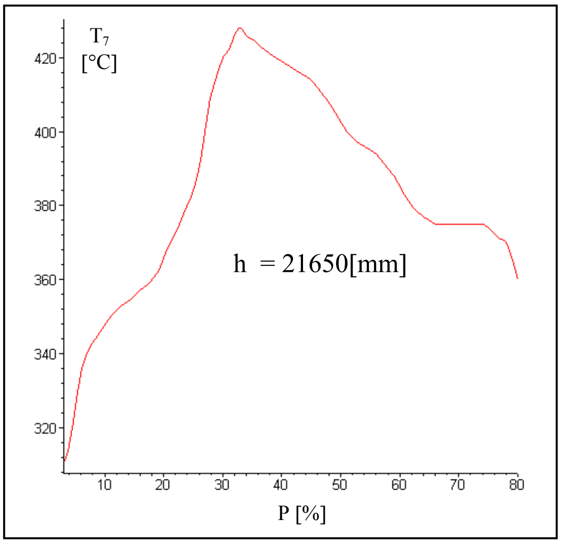
4.d)



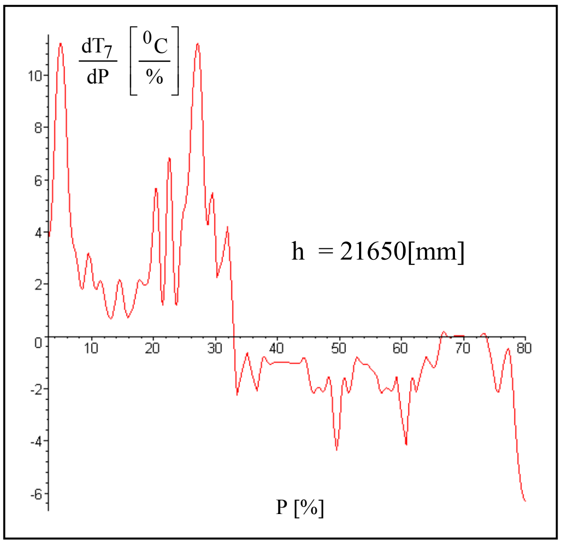
3.e)



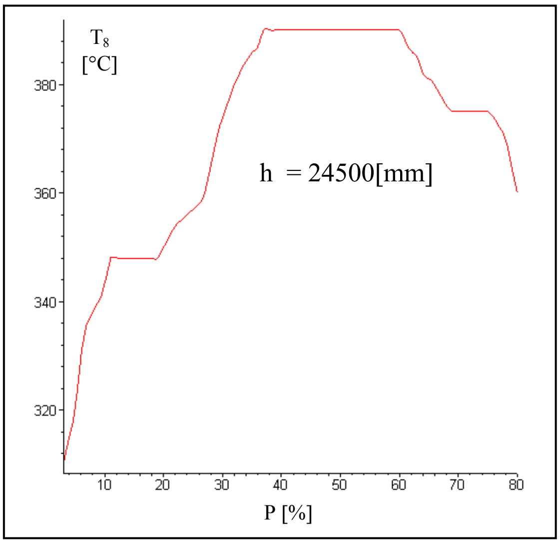
4.e)



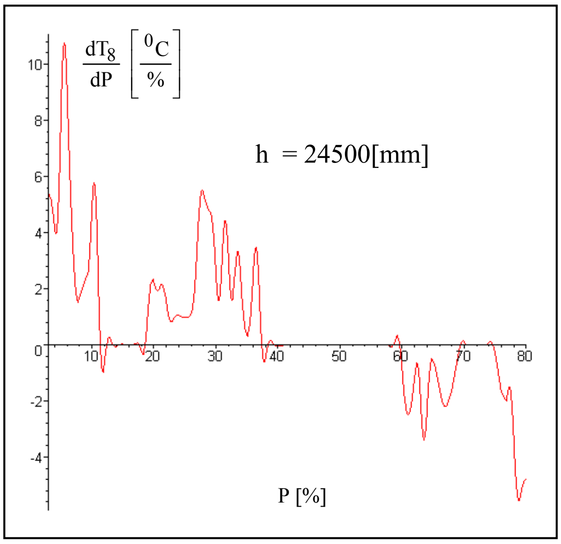
3.f)



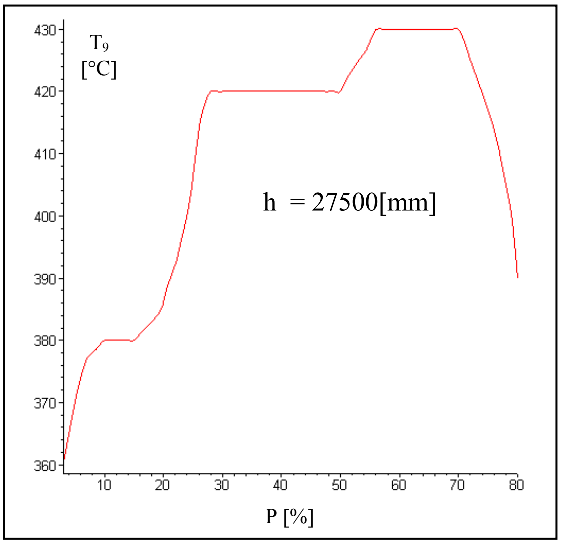
4.f)



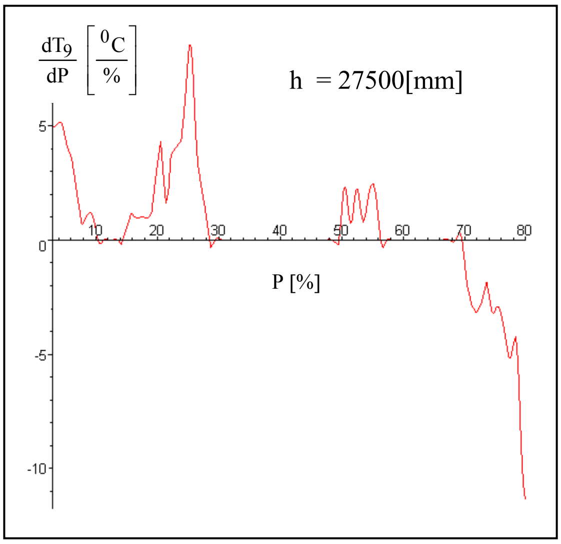
3.g)



4.g)



3.h)



4.h)

Figure 3: The function T(P, h=const) [the metallic wall's temperature T, function of the coking cycle relative duration P and the dimension h=const.] (separate plotting)

Figure 4: The gradient of the temperature's variation speed function of the relative duration of the coking cycle, P, at the dimension h=const. (separate plotting)

In figures 5 and 6 this graphs are plotted superimposed.

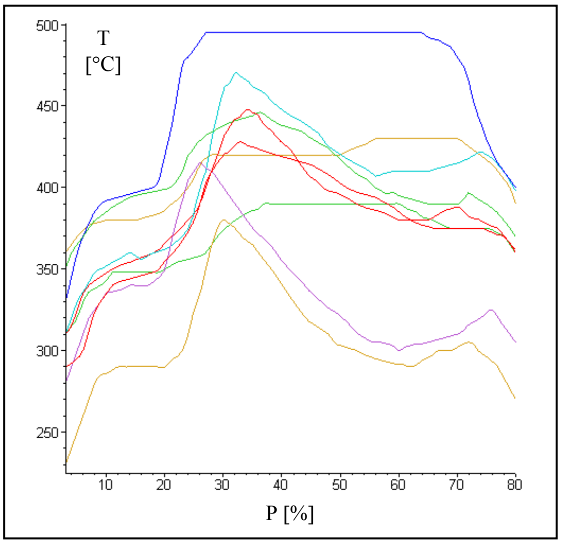


Figure 5: The function T(P, h=const) [the metallic wall's temperature T, function of the coking cycle relative duration P and the dimension h=const.] (superimposed plotting)

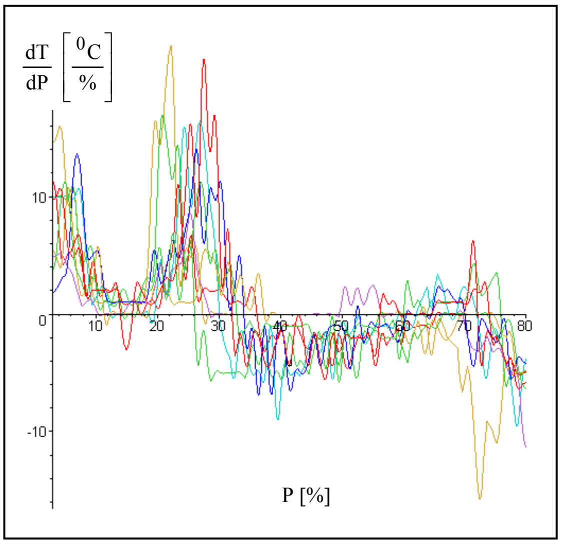


Figure 6: The gradient of the temperature's variation speed function of the relative duration of the coking cycle, P, at the dimension h=const. (superimposed plotting)

4. CONCLUSIONS

The coking chamber has been considered the best example for the creep phenomenon's study, associated with the thermic-oligo-cyclic fatigue.

The most important input data for the Finite Element Method programs, in this case of a coking chamber, is the value of the temperature in the metallic wall, T, which is distributed depending on the chamber's height h and the time percent of the coking cycle P. So, it was conceived a calculus program using the interpolation method with the third order Spline Functions, which established the numerical expression T(P,h) function, and it is recognized by the program in it's mathematical form.

BIBLIOGRAPHY

1. Nicoleta Rizea „Contributii la calculul de fluaj si oboseala al invelisurilor subtiri cu aplicatie la aparatura petrochimica" teza de doctorat, 18 aprilie 2008
2. Nicoleta Rizea: „Generarea corpurilor solide cu geometrie complexă în 3D" - în volumul Simpozionului International „Grafica 2000", Craiova, 19 – 21 octombrie 2000, pag.973-976
3. * * * Proiect I.C.I.T.P.R. –Camera de cocsare
4. * * * ASME Code. Case N-47: Design analysis of components in elevated temperature service" ed. 1999
5. * * * Criteria for Design of Elevated Temperature, Clas 1 Components in Section III, Division 1, of the ASME Boiler and Pressure Vessel Code ASME
6. * * * Maple 11, Maplesoft a Division of Waterloo Maple Inc 2007
7. * * * CosmosM 2007, Structural Research and Analysis Corporation 2007 (SRAC)
8. * * * Autocad 2007, Autocad Inc.2007
9. * * * SolidWorks 2007, SolidWorks Inc.2007