BULETINUL Universităţii Petrol – Gaze din Ploieşti	Vol. LXIV No. 1/2012	55-64	Seria Tehnică

Calculus Program for Using the Assessment Procedure Fitness For Service API 579

Ion Pană, Mihail Minescu

Universitatea Petrol-Gaze din Ploieşti, Bd. Bucureşti 39, Ploieşti
e-mail: ion.pana@upg-ploiesti.ro, mminescu@upg-ploiesti.ro

Abstract

The paper presents a computer program for transforming the results of an inspection of an oil pipeline through the procedure fitness for service. The pipeline was evaluated by means of standard ASME B31 G by the company carrying out the inspection and the processing of results is made by the standard methods API 579, using a program developed in Matlab. This paper is a continuation of the article [1].

Key words: *fitness for service, Matlab program*

Introduction

In the paper [1] the description of a method of transforming the results of an oil pipeline inspection was made. The inspection results were initially evaluated by the method of standard ASME B 31G, using the software of the inspection company [5]. At the request of the beneficiary of the research contract [2], the transformation was done in specific assessments of standard API 579, using a computer program developed in Matlab. The results were compared using the maximum allowable working pressure defined by each of the two methods.

The Description of the Program Used

Further we present the computer program realized in Matlab, as a series of steps. Each step is accompanied by the necessary code written in Matlab language.

Step 1. We introduce the necessary data for calculation: *FCA* , *LOSS*, D_e, *p*, *SMYS*, c_s , S_a, E_l, E_c, *RSFa* and import the matrix of data *data1* with the structure from Table 2 [1]. The results are stored in the matrix of analysis *b1*. The matrix of analysis *b1* includes for each defect (a defect is characterized by a line in the data matrix, *data1*, Table 3 [1]) the following answers grouped by columns, Table 4 [1] (ex. b1 (i, 2) = 1 shows that the defect from line *i* is the type of general lack of material): if all preliminary checks for longitudinal defect are fulfilled; the size of corrosion defect; the verification concerning groove like flaw defect; maximum allowable working pressure; the longitudinal extension check; the working pressure if the longitudinal extension of the defect is unacceptable; the check of preliminary conditions for the circumferential extending of the defect; the verification of the circumferential extension of the defect; if the analized defects are forming a group.The corrosion defects are analyzed both for the general loss of material (GLM, [4] Section 4) and for the local loss of material (LTA local thin area, [4] Section 5).

```
% the analyze of pipeline defect
% future corrosion allowance applied to the region of local metal loss , [mm]
FCA=1;
% outside diameter of the pipe, [mm]
De=20*25.4;
% working pressure, [MPa]
p=6.4;
% flow stress, [MPa]
Smys=358;
% safety coefficient for pipe material,  [-]
cs=1.4;
%  Sa allowable stress determined based on the original construction code of component, [MPa
Sa=Smys/cs;
% El longitudinal weld joint efficiency, [-]
El=1;
% Ec circumferential weld joint efficiency, [-]
Ec=1;
% RSFa allowable remaining strength factor, [-]
RSFa=0.9;
% Data matrix , date1, it is introduced in the working space
% the dimension of data matrix m number of lines,n number of columns
[m,n]=size(date1);
% Analyze matrix b1
```

Step 2. We calculate the required values for the analysis. The nominal thickness t_n was considered the effective value measured t_{rd}, $t_n = t_{rd}$, from the fourth column of *data1*, data1 (i, 4). The inner diameter D_i, the inner radius R_i, the average radius R_m, the average thickness t_{am} and the minimum thickness t_{mm} are calculated:

$$t_{mm} = \left(1 - \frac{A_{mm}}{100}\right) t_{nom} \tag{1}$$

$$t_{am} = \left(1 - \frac{A_{am}}{100}\right) t_{nom} \tag{2}$$

where the minimum depth measured A_{mm} (expressed in percentage) corresponds to the column 8 of the matrix *data1* data1(i, 8) and the average depth measured A_{am} (expressed in percentage) corresponds to the column 9 of the matrix *data1* data1(i, 9). The width of fault w corresponds to the column 7 of the matrix *data1* data1(i, 7), the length of fault L corresponds to the column 6 of the matrix *data1* data1(i, 6).

```
for i=1:1:m
% tom nominal or furnished thickness of the component adjusted for mill under tolerance as applicable,
%[mm]
tnom= date1(i,4);
%Di inside diameter of the pipe, [mm]
  Di=De-2*tnom; Ri=Di/2;Rm=(Di+De)/4;  tam=(1-date1(i,9)/100)*tnom;
% tmm minimum thickness ,[mm]
tmm=(1-date1(i,8)/100)*tnom;
% s axial length of the defect, [mm]
  s=date1(i,6);
% c width of the defect, [mm]
  c=date1(i,7);
% tc corroded wall thickness away from the region of local metal loss, [mm]
tc=tnom-FCA;
```

Step 3. We determine the values for t_{min} ,t_{sl} ,t_c, F, R_t, λ_l according to the formulas of [4].

```
% factor of material YB31, -
 YB31=0.4;MA=0;  t_C_min=p*De/2/(Sa*Ec+YB31*p)+MA;
% axial force F, N
   F=pi*p*Di^2/4;
% tsl thickness required for supplemental loads, [mm];
  tsl=F/2/Sa/Ec/pi/Rm; t_L_min=p*De/4/(Sa*El+YB31*p)+tsl+MA;
  tmin=max(t_C_min,t_L_min);
% Rt remaining thickness ratio, [-]
  Rt=(tmm-FCA)/tc;
% lambda_l  longitudinal flaw length parameter [-]
  lambda_l=1.285*s/(Di*tc)^0.5;
```

Step 4. We check the conditions for a fault of type local lack of metal LTA (relations 8-10 [1]). The distance to the discontinuity L_{msd} was considered to welding / equipment / interventions performed to the pipe and is determined by columns 2 and 3 of matrix *data1* data1(i , 2), data1(i, 3).

```
% b(i,1) the result of veification 1 true; 0 false
% for LTA conditions, section 5
if (Rt>=0.2) && (tmm-FCA>= 2.5) && (1e3*abs(date1(i,2))>= 1.8*(Di*tc)^0.5)
     b1(i,1)=1;
else
     b1(i,1)=0;
end
```

Step 5. We characterize the size of the defect according to Table 3 [1], s. the comments on the column 2.

```
% The characterisation of defect type
% b1(i,2) can be
% b1(i,2)= 1 General  lost of material GENE , API 579, Section 4
% b1(i,2)= 2 local lost of material Pitting PITT, API 579, Section 5,  with LTA
% b1(i,2)= 3  Axial grooving AXGR , Section 5 with LTA c=gw,s=gl
% b1(i,2)= 4 Circumferential grooving CIGR, Section 5 with groove s=gw,c=gl
% gr =gw/2
% b1(i,2)= 5 Pinhole PINH
% b1(i,2)= 6 Axial slotting AXSL , API 579, Section 9
% b1(i,2)= 7 Circumferential slotting CISl , API 579, Section 9
% b1(i,2)= 8 Another case
if (tnom<10)
     A=10;
else
     A=tnom;
end
  W=c;L=s;
if (W>=3*A) && (L>=3*A)
     b1(i,2)=1;
elseif (W >= A) && (W < 6*A) &&  (L >= A) && (L < 6*A) && (L/W>0.5) ...
&& (L/W < 2) && ~((W>=3*A) && (L>=3*A))
     b1(i,2)=2;
elseif (W >= A) && (W < 3*A) && (L/W >= 2)
     b1(i,2)=3;
elseif (L/W <= 0.5) && (L >= A) && (L < 3*A)
     b1(i,2)=4;
elseif (W > 0) && (W < A) && (L > 0) && (L < A)
     b1(i,2)=5;
elseif (W > 0) && (W < A) && (L >= A)
```

```
    b1(i,2)=6;
elseif (L> 0) && (L < A) && (W >= A)
    b1(i,2)=7;
else
    b1(i,2)=8; % Another case
end
```

Step 6. If the fault is of the GLM type (general loss of material b1 (i, 2) = 1), we check according to section 4 from [4], with the equation (14) [1]. If the verification is accepted b1 (i, 5) = 1, otherwise b1 (i, 5) = 0. The additional verifications are expressed by conditions (12, 13) [1]. The check result is inserted into b1 (i, 3).

```
% b1(i,3) if groove is OK
% or the general lost of material is OK
if b1(i,2)==1
tlim=max(0.2*tnom,2.5);
if (tmm-FCA>= max(0.5*tmin,tlim))
        b1(i,3)=1;
else
        b1(i,3)=0;
end
end
```

Step 7. If you have a groove-like flaw in the area with local metal loss, it must be checked if the radius at the base of a groove-like flaw satisfies the condition (11) [1]. The radius at the base of a groove-like flaw was considered half of the width of groove (the only parameter available). Checks are made separately for axial grooving, b1 (i, 2) = 3 and circumferential grooving, b1 (i, 2) = 4. The result of the test is passed in b1 (i, 3), that for these types of defects has the above specified meaning.

```
if b1(i,2)==3
gw=c;   gr=gw/2;
if (gr >= (1-Rt)*tc)
        b1(i,3)=1;
else
        b1(i,3)=0;
end
end
if b1(i,2)==4
gw=s;  gr=gw/2;
if (gr >= (1-Rt)*tc)
        b1(i,3)=1;
else
        b1(i,3)=0;
end
end
```

Step 8. We calculate the maximum allowable working pressure stored in b1 (i, 4), according to [4].

```
MAWP_c=2*Sa*Ec*(tc-MA)/(De-2*YB31*(tc-MA));
MAWP_l=4*Sa*El*(tc-tsl-MA)/(De-4*YB31*(tc-tsl-MA));
MAWP=min(MAWP_c,MAWP_l);
% b(i,4) it is used for MAWP
 b1(i,4)=MAWP;
```

Step 9. With the values of R_t and λ_l we check if the longitudinal extension of the defect is acceptable (Fig. 1) from [1]. To perform this check, instead of Fig. 1 we use the function R_{tl} =f(λ) for λ_l value:

$$R_{t1} = \begin{cases} 0{,}2; \lambda \le 0{,}354 \\ \dfrac{RSFa - \frac{RSFa}{M_t}}{1 - \frac{RSFa}{M_t}} & ; 0{,}354 < \lambda \le 20 \\ 0{,}9; \lambda \ge 20 \end{cases} \tag{3}$$

If our value R_t at the same lambda is greater than or equal with R_{tl}, the check is accepted b1 (i, 5) = 1, otherwise b1 (i, 5) = 0. The calculation can be done if we have an LTA-type defect with groove and radius at the base of a groove-like flaw condition verified ((b1 (i, 3) = 1) and (b1 (i, 2) = 3)) or ((b1 (i, 3) = 1) and (b1 (i, 2) = 4)) or a simple defect type LTA (b1 (i, 2) = 2). In all the types of the LTA corrosion defects the check of formulas (8-10) [1] is required (b1 (i, 1) = 1).

```
if ((( b1(i,3)== 1) && (b1(i,2)==3))|| ( b1(i,2)== 2) ||   (( b1(i,3)== 1) && (b1(i,2)==4)))  && ...
(b1(i,1)==1) % if groove OK or  LTA
%  verification  figure  1, b1(i,5) the result of the verification with figure 1, 1 true; 0 false
if (Rt>= fig_5_6(lambda_l))        b1(i,5)=1;
else     b1(i,5)=0;
end
```

The function used is:

```
function Rt1 = fig_5_6(lambda)
RSFa=0.9;
if lambda <= 0.354      Rt1=0.2;
elseif lambda < 20     Rt1=(RSFa-RSFa/MT1(lambda))/(1-RSFa/MT1(lambda));
else
   Rt1=0.9;
end
```

Step 10. If the check of the longitudinal expansion of the defect (Fig. 1) [1] is not satisfied, the residual resistance coefficient *RSF* is calculated, using the Folias factor expression M_t and the maximum working pressure is recalculated, *MAWPr* [4]. The extent of the defect in the axial direction is accepted in a level 1 assessment at the pressure written in b1 (i, 6), in MPa. In the column 6 (in the case of the GML defect type) it is recorded the value of the recalculated pressure with the average thickness of the pipe minus the corrosion allowance t_{am} –*FCA*, [4].

```
function  [mt] = Mt(lambda1 )
%   factor Mt
if lambda1 <20
   lambda=lambda1;
else
lambda =20;
end
mt= 1.0010- 0.014195* lambda+ 0.29090*lambda ^2  -0.096420 *lambda^3 + 0.020890 ...
*lambda ^4 -0.0030540 *lambda ^5 +2.9570e-4* lambda^6 -1.8462e-5* lambda^7+  7.1553e-7...
*lambda ^8-1.5631e-8* lambda^9 +1.4656e-10* lambda^ 10;
end
```

```
% the calculus of RSF
if b1(i,5)==0
Mt=MT1(lambda_l); RSFl=Rt/(1-(1-Rt)/Mt);
if RSFl>=RSFa
          b1(i,6)=MAWP;
```

```
else
        b1(i,6)=MAWP*RSFl/RSFa;
end
end
% b1(i,6) the value of working pressure if the longitudinal extension % is not checked
```

Step 11. The preliminary calculus used to check the circumferential expansion. We calculate the parameter lambda on the circumferential direction λ_c. We calculate the coefficient of resistance on the circumferential direction *RSF*, using λ_c. We consider the conditions (15-19) [1]. The tensile strength factor *TSF* is determined from the relationship provided in [4]. The result of verification is b1 (i, 7) = 1 for all the checks accepted, b1 (i, 7) = 0 otherwise.

```
%b1(i,7) the result of the verification for circumferential extension
% 1 true; 0 false
  lambda_c=1.285*c/(Di*tc)^0.5; Mt=MT1(lambda_c);RSFc=Rt/(1-(1-Rt)/Mt);
  TSF=Ec/2/RSFc*(1+((4-3*El^2)^0.5)/El);
if (lambda_c <=9) && (Di/tc>=20) && (RSFc>=0.7) && (RSFc<=1) && (Ec>=0.7) && (Ec<=1) ...
&& (El >= 0.7) && (El <= 1)
      b1(i,7)=1;
else
      b1(i,7)=0;
end
```

Step 12. Here we check of the circumferential expansion of defect, using Fig. 2 of [1], with λ_C and R_t. The curves for inspection from Fig. 2 are described by function (4):

$$R_{t2} = \begin{cases} C_1 + \frac{C_2}{\lambda_c} + \frac{C_3}{\lambda_c^2} + \frac{C_4}{\lambda_c^3} + \frac{C_5}{\lambda_c^4} + \frac{C_6}{\lambda_c^5}, \lambda_{c-0.2} < \lambda_c \leq 9 \\ 0.2, \lambda_c \leq \lambda_{c-0.2} \end{cases} \quad (4)$$

The coefficients C_i, i=1,6 are presented in [4] and introduced in the following function *valoare*. The checks are above the curve (in the acceptance region), so it was considered covering for the intermediate cases, the next higher value. Thus, for a check at *TSL* = 0.65, we must be above the curve for 0.7 (for safety reasons). The circumferential extension of the defect is acceptable if the calculated R_t is greater than the value offered by this function, b1(i, 8) = 1, otherwise b1 (i, 8) = 0.

```
function valoare = fig_5_8(lambdac1,TSF1)
if (TSF1 <= 0.7)
if (lambdac1 <= 0.21)
            valoare = 0.2;
else
            valoare = 0.99221 - 0.11959 / lambdac1 - 0.057333 / lambdac1 ^ 2 + 0.016948 / lambdac1 ^ 3
- 0.0017976 / lambdac1 ^ 4 + 0.000069114 / lambdac1 ^ 5;
end;
elseif (TSF1 <= 0.75)
if (lambdac1 <= 0.48)
            valoare = 0.2;
else
            valoare = 0.96801 - 0.2378 / lambdac1 - 0.32678 / lambdac1 ^ 2 + 0.20684 / lambdac1 ^ 3 -
0.046537 / lambdac1 ^ 4 + 0.0039436 / lambdac1 ^ 5;
end;
elseif (TSF1 <= 0.8)
if (lambdac1 <= 0.67)
            valoare = 0.2;
else
            valoare = 0.94413 - 0.31256 / lambdac1 - 0.69968 / lambdac1 ^ 2 + 0.6502 / lambdac1 ^ 3 -
0.22102 / lambdac1 ^ 4 + 0.028799 / lambdac1 ^ 5;
```

```
end;

elseif (TSF1 <= 0.9 )
if (lambdac1 <= 0.98)
        valoare = 0.2;
else
        valoare = 0.89962 - 0.3886 / lambdac1 - 1.6485 / lambdac1 ^ 2 + 2.3445 / lambdac1 ^ 3 -
1.2534 / lambdac1 ^ 4 + 0.25331 / lambdac1 ^ 5;
end;
elseif (TSF1 <= 1.0)
if (lambdac1 <= 1.23)
        valoare = 0.2;
else
        valoare = 0.85947 - 0.40012 / lambdac1 - 2.7979 / lambdac1 ^ 2 + 5.0729 / lambdac1 ^ 3 -
3.5217 / lambdac1 ^ 4 + 0.91877 / lambdac1 ^ 5;
end;
elseif (TSF1 <= 1.2)
if (lambdac1 <= 1.66)
        valoare = 0.2;
else
        valoare = 0.78654 - 0.25322 / lambdac1 - 5.7982 / lambdac1 ^ 2 + 13.858 / lambdac1 ^ 3 -
13.118 / lambdac1 ^ 4 + 4.6436 / lambdac1 ^ 5;
end;
elseif (TSF1 <= 1.40)
if (lambdac1 <= 2.03 )
        valoare = 0.2;
else
        valoare = 0.72335 + 0.011528 / lambdac1 - 9.3536 / lambdac1 ^ 2 + 26.031 / lambdac1 ^ 3 -
29.372 / lambdac1 ^ 4 + 12.387 / lambdac1 ^ 5;
end;
elseif (TSF1 <= 1.8 )
if ( lambdac1 <= 2.66)
        valoare = 0.2;
else
        valoare = 0.60737 + 0.93796 / lambdac1 - 19.239 / lambdac1 ^ 2 + 64.267 / lambdac1 ^ 3 -
91.307 / lambdac1 ^ 4 + 48.962 / lambdac1 ^ 5;
end;
elseif (TSF1 <= 2.3)
if (lambdac1 <= 3.35)
        valoare = 0.2;
else
        valoare = 0.49304 + 2.1692 / lambdac1 - 32.459 / lambdac1 ^ 2 + 122.45 / lambdac1 ^ 3 -
202.43 / lambdac1 ^ 4 + 127.27 / lambdac1 ^ 5;
end;
else
      valoare = 0.2;
end;
end
```

Step 13. For the general lack of material defect type we use as a criterion for acceptance the average thickness (relation 14 [1]). In this case we put b1(i, 5)=1, if the acceptance criteria is checked, otherwise b1(i, 5)=0; b1 (i, 1) has no meaning, we put b1 (i, 1)=0, the result of the conditions for extending circumferential b1 (i, 7) is not necessary, b1 (i, 7)=0; the result of circumferential extend evaluation b1 (i, 8) is not necessary, b1 (i, 8)=0; in b1 (i, 6) we record the value of the recalculated pressure with the average thickness minus the corrosion allowance.

```
if (Rt>= fig_5_8(lambda_c,TSF)) && (b1(i,7)==1)
```

```
    b1(i,8)=1;
else
    b1(i,8)=0;
end
end % if groove OK or LTA
if (b1(i,2)==1) && (tam-FCA>= tmin) % general lost of material PGM
    b1(i,5)=1;    b1(i,1)=0;    b1(i,7)=0;    b1(i,8)=0;
    MAWP_cr=2*Sa*Ec*(tam-FCA-MA)/(De-2*YB31*(tam-FCA-MA));
    MAWP_lr=4*Sa*El*(tam-MA-FCA-tsl)/(De-4*YB31*(tam-FCA-MA-tsl));
MAWPr=min(MAWP_cr,MAWP_lr);
    b1(i,6)=MAWPr;
end %  general lost of material
end %  for
```

Step 14. It is assessed whether we have grouped defects. If defects are associated, forming a group then: b1 (i, 9) = 1 defect is part of a group, otherwise b1 (i, 9) = 0 defect is not part of a group. Rule of interaction of defects is given in Table 4 with references to column 9 [1]. For this we compute the difference between the positions of the defects on the pipe axis data column 1 of matrix *data1* data1 (i, 1), called $dist_L$ and compare it with the minimum length of the two defects involved, column 6 of the matrix data, *data1*. We compute the difference between the positions of the defects on the cross section (per hour) of pipe $dist_C$ column 5 of data matrix, *data1* and compare it with the minimum width of two defects involved in the matrix column 7 data, *data1*. For that we transform the clock's positions in lengths on the unfolded pipe:

$$dist_L = 1e3 \cdot |date1(i,1) - date1(i+1,1)| \tag{5,a}$$

$$dist_C = \left|[date1\,(i,5)] + \frac{\{date1(i,5)\}}{60} - [date1(i+1,5)] - \frac{\{date1(i+1,5)\}}{60}\right| \cdot \frac{\pi}{6} \cdot R_m \tag{5,b}$$

and by using the condition of belonging to a group of defects:

$$if\left(dist_L < \min\left(date1(i,6), date1(i+1,6)\right)\right) or \left(dist_C < \min\left(date1(i,7), date1(i+1,7)\right)\right)$$
$$b1(i,9) = 1 \quad \text{and} \quad b1(i+1,9) = 1 \tag{6}$$

```
% the group of defects
for i=1:1:m-1
dist_L=1e3*abs(date1(i,1)-date1(i+1,1));
piA=floor(date1(i,5)); pfA=date1(i,5)-piA;
piB=floor(date1(i+1,5));
pfB=date1(i+1,5)-piB;
dist_C=abs(piA+pfA/60-piB-pfB/60)*pi/6*Rm;
if (dist_L < min(date1(i,6),date1(i+1,6))) || ( dist_C < min( date1(i,7),date1(i+1,7) ))
    b1(i,9)=1;    b1(i+1,9)=1;
end
end
```

Step 15. Final analysis, the result of the program execution is the matrix of analysis *b1* was shown in Table 5 [1]. The values calculated in Matlab are downloaded next into the initial Excel files, as new columns. A picture of the provided data with the new values calculated according the API 579 procedure and elements of comparison (last two columns) was given in Table 6 [1].

Conclusions

The main conclusions about the use of the program were made in [1]. Referring strictly to the technique of the implementation of the program, the use of Matlab application has the following advantages:

- allows the manipulation of the large matrices;
- the using of the Matlab statistical calculation tools;
- allows the easy importing into Excel and the comparison of the results.

In terms of flexibility the method has the disadvantage of the lack of a dedicated graphical interface. This can be achieved by the design of such interface with a database connection as in [3], but the cost of the achieving of such software are great, and cannot be justified considering the actual reduction of pipeline transport activities of the involved companies.

List of Notations

c – maximum depth of the region of local metal loss [mm]
c_s – safety coefficient for pipe material [-]
D_e – outside diameter of the pipe [mm]
D_i – inside diameter of the pipe [mm]
E_c – circumferential weld joint efficiency [-]
E_l – longitudinal weld joint efficiency [-]
ERF – estimated repair factor, *ERF = MAWP/ Psafe* [-]
F – applied net-section axial force for the weight [N]
FCA – future corrosion allowance applied to the region of local metal loss [mm]
L – axial length of the defect [mm]
L_{msd} – distance to the nearest major structural discontinuity [mm]
LOSS – amount of uniform metal loss away from the local metal loss location at the time of the assessment [mm]
M_i – bulging stress magnification factor, i = 1,2 [-]
MWP – maximum working pressure [MPa]
MAWP – maximum allowable working pressure [MPa]
MAWPr – reduced permissible maximum allowable working pressure [MPa]
M_t – Folias factor [-]
p – working pressure [MPa]
P_{design} – design pressure (safety factor 0.72) [MPa]
PO – operating pressure, may equal *MAWP* or *MWP* [MPa]
P_{safe} – safe operating pressure [MPa]
R_i – inside radius of pipe [mm]
R_m – average radius of the pipe [mm]
RSFa – allowable remaining strength factor [-]
RSF – computed remaining strength factor based on the meridional extent of the LTA [-]
R_t – remaining thickness ratio [-]
S_a – allowable stress determined based on the original construction code of component [MPa]
SF – estimated failure stress level [MPa]
S_{flow} – flow stress [MPa]
S_O – hoop stress at the operating pressure, calculated as $P_O D/2t_n$ [MPa]
SF – safety factor [-]
t_n – nominal wall thickness [mm]
t_{am} – mean wall thickness measured [mm]
t_c – corroded wall thickness away from the region of local metal loss [mm]
t_{min} – minimum required thickness for the component that governs the *MAWP* calculation [mm].

References

1. Minescu M., Pană I. – The equivalence of the assessment procedures API 579 and ASME B31G, *Buletinul Universității Petrol – Gaze din Ploieşti*, *Seria tehnică,* Vol. LXIV, No. 1/2012, pp. 11-20.
2. Minescu M., a.o. – *Research service contract No. 387/09.12.2010 (UPG No. 58/21.12.2010) "Implementation of an assessment procedure type Fitness - For - Service in accordance with API*

Standard 579, for oil-ducts faults and technical assessment of a real pipe section", beneficiary S.C. Conpet SA, Petroleum-Gas University of Ploiesti 2011.
3. P a n ă , I . – Evaluarea abaterii de la circularitate a conductelor conform standardului API 579 Apt pentru serviciu, *Lucrările sesiunii de comunicări ştiinţifice a Catedrei de Mecanică Tehnică şi Mecanisme*, Universitatea Tehnică de Construcţii Bucureşti, SIMEC 2011, Editura Matrix Rom Bucureşti, 2011, pp. 169-174.
4. * * * – *API 579 / ASME FFS-1, Fitness-For-Service*, Washington, 2007.
5. * * * – *Final Report of Inspection: Inspection geometric and Metal Loss, Project Number 3-4000-10286 Rosen, CONPET Oil Pipeline 20 "* , Section Constanta-C01, October / November 2008.

Program de calcul pentru utilizarea procedurii de evaluare “Fitness For Service” API 579

Rezumat

Articolul prezintă un program de calcul care permite transformarea rezultatelor unei inspecţii de la o conductă de petrol, prin procedura apt pentru serviciu. Conducta a fost evaluată prin metoda din standardul ASME B31 G de către firma care a efectuat inspecţia şi transformarea rezultatelor este făcută prin metodele standardului API 579, utilizînd un program realizat în Matlab. Lucrarea este o continuare a articolului [1].