WYZNACZANIE PARAMETRÓW KOROZYJNYCH ZA POMOCĄ ZMODYFIKOWANEJ METODY PRZESZUKIWANIA PRZESTRZENI PARAMETRÓW

IGOR SKALSKI
skalgo@ship.cto.gda.pl

Centrum Techniki Okrętowej, Gdańsk
Ośrodek Materiałoznawstwa, Korozji i Ochrony Środowiska

W pracy przedstawiono zmodyfikowaną metodę dopasowania krzywej teoretycznej do punktów eksperymentalnych za pomocą przeszukiwania przestrzeni parametrów. Przedstawiono również kompletny program napisany w języku C (w standardzie ANSI C) wyznaczający parametry korozyjne dla procesu przebiegającego z kontrolą aktywacyjną.
Po dokonaniu niewielkich zmian program może posłużyć do innych celów.

1. Wprowadzenie

W pracy przedstawiono kompletny kod źródłowy procedury numerycznej w języku C, w standardzie ANSI C [1][2][3], pozwalającej na dopasowanie krzywej teoretycznej opisującej proces korozyjny przebiegajacy z kontrolą aktywacyjną do punktów doświadczalnych otrzymanych podczas badań polaryzacyjnych. Procedura wykorzystuje znaną metodę [4] przeszukiwania przestrzeni parametrów. Istotnym uzupełnieniem, znacznie zwiększającym zbieżność obliczeń, jest wprowadzona w procedurze korekcja wartości kroku dla parametrów wpływających na uzyskiwane wyniki.

2. Podstawy teoretyczne

Podczas polaryzacji potencjodynamicznej otrzymujemy szereg par punktów przedstawiających w sposób dyskretny funkcję:

$$I_p = f(E_p) \tag{1}$$

gdzie:

E_p - wartość narzuconego potencjału
I_p - wartość odpowiedzi prądowej

Zjawisko korozyjne przebiegające z kontrolą aktywacyjną, w którym zasadniczą rolę odgrywa jeden proces elektrochemiczny można z pewnym przybliżeniem opisać równaniem [5]

$$I_p = I_c\left\{exp\frac{2.303(E_p - E_c)}{b_a} - exp\frac{-2.303(E_p - E_c)}{b_c}\right\} \tag{2}$$

gdzie:

E_c - potencjał korozyjny
I_c - prąd korozyjny
b_a - anodowy współczynnik Tafela
b_c - katodowy współczynnik Tafela

Przedstawiona procedura przeszukuje przestrzeń parametrów określających funkcję

$$I_p = f(E_p, E_c, I_c, b_a, b_c) \tag{3}$$

dopasowując tę funkcję do punktów doświadczalnych. Miarą dopasowania krzywej teoretycznej do punktów jest wyrażenie:

$$S_{(E_c,I_c,b_a,b_c)} = \sum_{i=1}^{n}\left(f(E_i, E_c, I_c, b_a, b_c) - I_i\right)^2 \tag{4}$$

przedstawiające sumę kwadratów odległości punktów od dopasowywanej krzywej.

Minimalizowanie wartości $S_{(E_c,I_c,b_a,b_c)}$ polega na cyklicznej korekcji wartości parametrów E_c, I_c, b_a i b_c dokonywanej w oparciu o wyniki obliczeń wartości elementów macierzy

$$m_{i,j,k,l} = S(E_c + (i-1)\Delta E_p, I_c + (j-1)\Delta I_p, b_a + (k-1)\Delta b_a, b_c + (l-1)\Delta b_c) \tag{5}$$

gdzie:

$i, j, k, l = \{0, 1, 2\}$
ΔE_p, ΔI_p, Δb_a, Δb_c - kroki dopasowania

i tak: jeżeli najmniejszą wartość S otrzymuje się dla elementu $m_{1,1,1,1}$, to następuje zmniejszenie kroków dopasowania ΔE_p, ΔI_p, Δb_a i Δb_c. W przeciwnym przypadku następuje zmiana wartości parametrów E_c, I_c, b_a i b_c na te, które umożliwiły uzyskanie najmniejszej wartości $m_{i,j,k,l}$ oraz zwiększenie wartości tych kroków dopasowania, które wpłynęły na zmniejszenie wartości $m_{i,j,k,l}$. Wielokrotne zmniejszenie kroków dopasowania nie prowadzące do zmniejszenia wartości S oznacza osiągnięcie najlepszego dopasowania.

3. Implementacja

W celu uzyskania dużej przenośności, procedura została napisana w języku C, w standardzie ANSI C. Próby przeprowadzono na komputerach: PC 386 DX 40 z koprocesorem arytmetycznym oraz

na PC 586 133. Obydwa komputery były wyposażone w systemy operacyjne: *UNIX*[1] (*Linux* z jądrem v. 2.0.29) i *DOS*. Program był kompilowany za pomocą kompilatora *gcc*[2] (pod *UNIX*'em) oraz *gcc* (*DJGPP*) i *Turbo C* v. 2.0 pod *DOS*'em.

Dokonano obliczeń dla różnego rodzaju danych symulowanych komputerowo oraz zarejestrowanych podczas badań polaryzacyjnych. Wyniki porównywano w wynikami otrzymanymi dla tych samych danych za pomocą jednego z programów komercyjnych, dokonującego obliczeń za pomocą procedury Marquardta. Zazwyczaj otrzymywano identyczne wyniki, czasami udawało się za pomocą przedstawionego w niniejszej publikacji programu uzyskać nieco lepsze dopasowanie krzywej.

4. Kod źródłowy programu

Niżej przedstawiono kompletny kod programu[3] wyznaczającego parametry korozyjne za pomocą zmodyfikowanej metody przeszukiwania przestrzeni parametrów dla przykładowych danych[4].

Uwaga: W języku C nie stosuje się numeracji linii programu. Numery linii na wydruku mają ułatwić jego komentowanie i należy je opuścić podczas wprowadzania kodu do komputera.

```
 1  #include<stdio.h>
 2  #include<stdlib.h>
 3  #include<math.h>
 4  #include<float.h>
 5
 6  #define mn 100
 7  int pn;
 8  float ep[mn],ip[mn],qsum,
 9        icorr,ecorr,ba,bc,
10        d_icorr,d_ecorr,d_ba,d_bc,
11        t_icorr,t_ecorr,t_ba,t_bc,
12        m_icorr,m_ecorr,m_ba,m_bc;
13
14  void results(float icorr,float ecorr,float ba,float bc,float qsum)
15  {
16    printf("\nicorr: %15.3f\necorr: %15.1f",icorr,ecorr);
17    printf("\nba   : %15.1f\nbc   : %15.1f\nqsum : %15g\n",ba,bc,qsum);
18  }
19
20  float sqdif()
21  {
22    int i;
23    float s=0,t;
24    for(i=0;i<pn;i++)
25    {
26      t=t_icorr*((exp((2.303*(ep[i]-t_ecorr))/t_ba)
27                 -exp((-2.303*(ep[i]-t_ecorr))/t_bc)));
28      s+=(t-ip[i])*(t-ip[i]);
```

[1] "UNIX" jest nazwą zastrzeżoną AT&T Bell Laboratories.

[2] za pomocą instrukcji: `gcc -ansi -pedantic -Wall -lm fit.c -o fit`

[3] W efekcie działania programu otrzymano: icorr: 19.132, ecorr: -633.4, ba: 35.4, bc: 71.3, qsum: 1.55476

[4] Dane te zostały zarejestrowane podczas polaryzacji stali węglowej St4S eksponowanej 70 minut w 3 % roztworze NaCl intensywnie mieszanym za pomocą mieszadła magnetycznego. Stosowano szybkość przesuwu potencjału 10 mV/min. W celu ograniczenia objętości pracy z zarejestrowanych danych wybrano jedynie kilka wartości.

```
29    }
30    return(s);
31  }
32
33  void cfit()
34  {
35    int i,j,k,l,mi=1,mj=1,mk=1,ml=1,c=0,cp;
36    float s;
37
38    icorr=10;ecorr=-600;ba=120;bc=120;
39    d_icorr=.1;d_ecorr=.1;d_ba=.1;d_bc=.1;
40
41    while(c<100)
42    {
43      m_icorr=icorr;m_ecorr=ecorr;m_ba=ba;m_bc=bc;cp=0;
44      for(i=0;i<3;i++)
45        for(j=0;j<3;j++)
46          for(k=0;k<3;k++)
47            for(l=0;l<3;l++)
48            {
49              t_icorr=m_icorr+(i-1)*d_icorr;
50              t_ecorr=m_ecorr+(j-1)*d_ecorr;
51              t_ba=m_ba+(k-1)*d_ba;
52              t_bc=m_bc+(l-1)*d_bc;
53              if((s=sqdif())<qsum)
54              {
55                c=0;cp=1;qsum=s;
56                icorr=t_icorr;ecorr=t_ecorr;ba=t_ba;bc=t_bc;
57                mi=i;mj=j;mk=k;ml=l;
58              }
59            }
60      if(cp)
61      {
62        if(mi!=1)d_icorr*=1.1;if(mj!=1)d_ecorr*=1.1;
63        if(mk!=1)d_ba*=1.1;if(ml!=1)d_bc*=1.1;
64      }
65      else
66      {
67        c++;
68        d_icorr*=.9;d_ecorr*=.9;d_ba*=.9;d_bc*=.9;
69      }
70    }
71  }
72
73  void main()
74  {
75    qsum=FLT_MAX;
76    pn=6;
77    ip[0]=-37;        ep[0]=-657;
78    ip[1]=-24.75;     ep[1]=-649;
79    ip[2]=-12.3125;   ep[2]=-641;
80    ip[3]=-0.23975;   ep[3]=-633;
81    ip[4]=19.0125;    ep[4]=-625;
82    ip[5]=44.125;     ep[5]=-617;
83    cfit();
84    results(icorr,ecorr,ba,bc,qsum);
85  }
```

5. Opis programu

Po przyłączeniu w liniach 1÷4 bibliotek, z których korzysta program, w linii 6 występuję deklaracja stałej określającej maksymalną liczbę punktów, jaką można wprowadzić do programu. Wpisana wartość jest przykładowa i nic nie stoi na przeszkodzie aby ją zwiększyć. Kolejne linie

7÷12 zawierają deklaracje zmiennych globalnych, przy czym w zmiennej typu całkowitego pn będzie przechowywana informacja o liczbie wprowadzonych par punktów. Zmienna typu zmiennoprzecinkowego qsum w linii 75 jest inicjowana maksymalną wartością obsługiwaną przez język C w danym systemie operacyjnym. Po zakończeniu działania funkcji cfit() zmienna ta będzie zawierała osiągniętą minimalną wartość sumy kwadratów odchyleń punktów od dopasowanej krzywej teoretycznej.

Wykonanie programu rozpoczyna się od wywołania funkcji main() (w linii 73). Po zainicjowaniu zmiennej pn i tablic, zostaje wywołana funkcja cfit(). W linii 38 odpowiednie zmienne są inicjowane wartościami początkowymi, a w linii następnej ustalane są wartości kroków dopasowania. W liniach 41÷70 zawarto pętle programowe oraz instrukcje decyzyjne. Wyjście z głównej pętli while nastąpi, gdy stukrotne zmniejszenie wartości kroków dopasowania nie doprowadzi do zmniejszenia wartości sumy kwadratów odchyleń (zmiennej qsum) od dopasowywanej krzywej. Obliczanie sumy kwadratów ochyleń następuje w funkcji sqdif() zawartej w liniach 20÷31 i wywoływanej w linii 53. W funkcji sqdif() zawarto równanie (2).

Na koniec w funkcji main() zostaje wywołana funkcja results() wyświetlająca wyniki obliczeń.

6. Wnioski

Przedstawiona procedura charakteryzuje się prostotą i jest zapisana w postaci stosunkowo krótkiego kodu. Dzieje się tak kosztem sporej mocy obliczeniowej koniecznej do uzyskania wyników. Procedura nadaje się głównie do dokonywania obliczeń dla niewielkiej liczby punktów pomiarowych. Niezbędnym minimum (przy przetwarzaniu danych zawierających kilkanaście punktów pomiarowych) wydaje sie być PC 386 DX 40 z koprocesorem arytmetycznym, gdzie uzyskuje się wyniki w czasie od kilkunastu do kilkudziesięciu sekund.

Otrzymane wyniki – tak jak w przypadku zastosowania podobnych metod numerycznych – są zależne od wartości początkowych parametrów E_c, I_c, b_a i b_c oraz od początkowych wartości kroków dopasowania ΔE_p, ΔI_p, Δb_a i Δb_c. Szczególnie istotne jest podanie zbliżonej do rzeczywistej wartości E_c.

Stosowanie programu w praktyce wymaga jego uzupełnienia o funkcję odczytu danych z dysku, funkcję wyznaczania wartości E_c przed rozpoczęciem obliczeń oraz szacowanie błędu dopasowania krzywej. Przydatnym byłoby również umożliwienie użytkownikowi programu ręcznego wprowadzenia danych startowych i kroków dopasowania oraz podglądu graficznej reprezentacji danych i dopasowanej krzywej.

7. Literatura

[1] "Język ANSI C", Kernighan, B. W., Ritchie, D. M., Wydawnictwa Naukowo – Techniczne, Warszawa, 1997.

[2] "ANSI C for programmers on UNIX Systems", Love, T., Publikacja dostępna w sieci Internet.

[3] "Notes on Writing Portable Programs in C", Dolenc, A., Lemmke, A., Keppel, D., Reilly, G. V., Publikacja dostępna w sieci Internet.

[4] "2^5 numerycznych programów w języku BASIC", Tatarkiewicz, J., Witkowski, A., NOT – SIGMA, Warszawa 1987, s. 32

[5] Wagner, C., Traud, W., Z.Electrochem., Vol.44, p.391, 1938

■