Metoda Aitkena

Teoria:
Załóżmy, że algorytm:

$$x^{(k+1)} = \phi\big(x^{(k)}\big) \qquad k \geq 0 \tag{1}$$

zbiega liniowo do punktu stałego α funkcji ϕ , to dla ustalonego k mozemy zapisać, że

$$\phi\big(x^{(k)}\big) - \alpha = \lambda\big(x^{(k)} - \alpha\big) \tag{2}$$

Wzór ten możemy przekształcić tak, aby móc wyliczyć z niego punkt stały:

$$\alpha = \frac{\phi\big(x^{(k)}\big) - \lambda x^{(k)}}{1-\lambda} = \frac{\phi\big(x^{(k)}\big) - \lambda x^{(k)} + x^{(k)} - x^{(k)}}{1-\lambda}$$

i ostatecznie:

$$\alpha = x^{(k)} + \frac{\phi\big(x^{(k)}\big) - x^{(k)}}{1-\lambda} \tag{3}$$

Teraz należy wyznaczyć nieznaną wartość λ. W tym celu konstruujemy następujący ciąg:

$$\lambda^{(k)} = \frac{\phi\Big(\phi\big(x^{(k)}\big)\Big) - \phi\big(x^{(k)}\big)}{\phi\big(x^{(k)}\big) - x^{(k)}} \tag{4}$$

Lemat 1 *Jeżeli wzór iteracyjny (1) zbiega do punktu stałego α, to:*

$$\lim_{k\to\infty} \lambda^{(k)} = \phi'(\alpha) \tag{5}$$

Na podstawie Lematu 1 możemy przyjać, że dla danego k, $\lambda^{(k)}$ przybliża wartość λ we wzorze (3). Dzięki temu możemy zapisać nowy wzór iteracyjny:

$$x^{(k+1)} = x^{(k)} - \frac{\Big(\phi\big(x^{(k)}\big) - x^{(k)}\Big)^2}{\phi\Big(\phi\big(x^{(k)}\big)\Big) - 2\phi\big(x^{(k)}\big) + x^{(k)}} \qquad k \geq 0 \tag{6}$$

Zatem nowa funkcja w metodzie punktu stałego wyraża się wzorem:

$$F(x^{(k)}) = \frac{x^{(k)}\phi\Big(\phi\big(x^{(k)}\big)\Big) - \big[\phi\big(x^{(k)}\big)\big]^2}{\phi\Big(\phi\big(x^{(k)}\big)\Big) - 2\phi\big(x^{(k)}\big) + x^{(k)}} \tag{7}$$

Twierdzenie 1 *Zakładając, że $\phi(x) = x - f(x)$ i używając metody punktu stałego do obliczania zer funkcji f, to:*

- *jeżeli metoda punktu stałego zbiega się liniowo do pojedyńczego zera funkcji f, to metoda Aitkena zbiega się kwadratowo do tego samego zera;*
- *jeżeli rząd zbieżności metody punktu stałego wynosi $p \geq 2$ i zbiega się do pojedyńczego zera funkcji f, to metoda Aitkena zbiega się do tego samego zera i ma rząd zbieżności $2p - 1$;*
- *jeżeli metoda punktu stałego zbiega się liniowo do zera o wielokrotności $m \geq 2$, to metoda Aitkena zbiega się liniowo do tego samego zera ze współczynnikiem zbieżności asymptotycznej równym $C = 1 - 1/m$.*

Dodatkowo, jeżeli $p = 1$ i zero jest pojedyńcze, to metoda Aitkena zbiega do tego zera nawet jeżeli metoda punktu stałego dla tych samych danych jest rozbieżna.

Przygotował: Andrzej Kosior

Skrypt 1:

```
function y = funfix(x)

y = x - (.25 * x^2 - sin(x));
```

Skrypt 2:

```
function y = fixed_point_aitken ( f, x0, TOL, Nmax )

%FIXED_POINT_AITKEN   uzywa metody punktu stalego wraz z ekstrapolacja Aitkena
%                     do znalezienia zer rownan nieliniowych
%
%     wywolywanie:
%             y = fixed_point_aitken ( 'f', x0, TOL, Nmax )
%             fixed_point_aitken ( 'f', x0, TOL, Nmax )
%
%     dane wejsciowe:
%             f       (string containing) nazwa pliku m-file definiujacego
%                     funkcje, ktorejpunkt staly jest poszukiwany
%             x0      poczatkowe przyblizenie polozenia punktu stalego
%             TOL     odleglosc pomiedzy kolejnymi przyblizeniami bedaca
%                     warunkiem zatrzymania obliczen
%             Nmax    maksymalna liczba iteracji
%
%     dane wyjsciowe:
%             y       przyblizona wartosc punktu stalego

old = x0;   phatold = x0; for i = 1 : Nmax
    new = feval(f,old);

    if ( i == 1 | i == 2 )
       if ( nargout == 0 )
          disp ( sprintf ( '\t\t %3d \t %.10f \n', i, new ) )
       end
    else
       phat = new - ( new - old ) ^ 2 / ( new - 2 * old + older );

       if ( nargout == 0 )
          disp ( sprintf ( '\t\t %3d \t %.10f \t %.10f \n', i, new, phat ) )
       end

       if ( abs(phat-phatold) < TOL )
          if ( nargout == 1 )
             y = phat;
          end
          return
       else
          phatold = phat;
       end
    end
    older = old;
    old = new;
end
disp('Przekroczono maksymalna liczbe iteracji') if ( nargout == 1 )
y = phat;
end
```

Zadanie:

Wyznacz zero funkcji:

$$y = \frac{x^2}{4} + \sin x$$

w przedziale $[1.5, 2.0]$.

Rozwizanie w programie MATLAB:

```
clc
fixed_point_aitken ('funfix',1.5,1e-14,40)
y = fixed_point_aitken ('funfix',1.5,1e-14,40)
```