Pierwszy projekt z labu

Zaprogramować funkcję octave'a z testami ***metody Steffensena*** zdefiniowanej wzorem

$$x_{n+1} = x_n - \frac{f(x_n)^2}{f(x_n + f(x_n)) - f(x_n)}$$

- która ma znaleźć przybliżenie x^* rozwiązania $f(x^*) = 0$ oraz drukować na ekranie $|e_{n+1}|/|e_n|^p$ dla $p = 1, 2, 3$ i $e_n = x_n - x^*$. Jako warunek stopu przyjąć $|f(x_n)| < tola + tolr * |f(x_0)|$ lub to, że ilość iteracji przekroczyła zadaną liczbę $Nmax$.

Dokładnie napisać funkcję octave'a w m-pliku steff.m:

```
function [y,it]=steff(FCN, x0, sol, tolr, tola, Nmax)
```

z parametrami wejściowymi:

- FCN - 'wskaźnik' do funkcji f (function handle),
- $x0$ - przybliżenie startowe,
- sol - rozwiązanie równania x^*
- $tola, tolr$ - tolerancje błędu
- $Nmax$ - maksymalna ilość iteracji.

Funkcja ma zwracać:

- y - obliczone pierwiastek
- it - ilość iteracji.

W razie gdyby ilość iteracji przekroczyła $Nmax$ funkcja powinna zwrócić komunikat o tym na ekran.

Funkcja ma działać również przy podaniu tylko trzech lub czterech zmiennych, wtedy odpowiednie parametry mają przyjąć następujące wartości : $tola = tolr * 100 = 10^{-9}$ i $Nmax = 100$.

Przetestować rząd zbieżności tej metody dla dwóch funkcji $f_1(x) = x^k - 2 = 0$ dla $k = 2, 8$ i $f_2(x) = (x-2)^p = 0$ dla $p = 2, 8$ biorąc za x_0 różne dodatnie wartości np. $x_0 = 10^{-9}, 1, 2, 10^9$ czy $x^* + 10^{-7}$ (znamy przecież $x^* = 2^{1/k}$ lub $x^* = 2$ odpowiednio). Przy testowaniu mogę poprosić o inne równanie.