Metody numeryczne 2021/2022: lista 7

(*układy równań nieliniowych*)

1. Omów, przedstaw graficznie, i wyprowadź wzory iteracyjne dla metod:
   (a) bisekcji (połowienia, równego podziału),
   (b) Falsi,
   (c) siecznych,
   (d) Newtona (stycznych).

2. Jak można wykorzystać metodę Newtona do obliczania pierwiastka kwadratowego z liczby rzeczywistej? Zastosuj ten algorytm do obliczenia $\sqrt{2}$ z dokładnością do 3 miejsc znaczących.

3. (*równania nieliniowe i interpolacja odwrotna*) Niech funkcja $f(x)$, zadana na przedziale $\langle a, b\rangle$, ma funkcję odwrotną $g(y) \equiv f^{-1}(y)$. Zauważmy, że $x^* \equiv g(y = 0)$ jest pierwiastkiem równania $f(x) = 0$. Zapisz wzór na wielomian interpolacyjny Lagrange'a rzędu n, $W_n(x)$, dla funkcji $g(y)$ i znajdź przybliżoną wartość $x^* \approx W_n(y = 0)$. Czy któraś z metod z zad. 1. jest szczególną realizacją tej procedury?

4. Oszacuj z góry błąd metody Newtona jeżeli w otoczeniu pierwiastka równania $f(x) = 0$, dwie pierwsze pochodne są dodatnie ($f'(x) > 0$ i $f''(x) > 0$).

5. Uzasadnij, że funkcja $u(x) \equiv \frac{f(x)}{f'(x)}$ ma tylko pierwiastki jednokrotne. Jak można zastosować $u(x)$ do wyszukiwania pierwiastków wielokrotnych funkcji $f(x)$?

6. **(Zadanie numeryczne NUM 9)** Znajdź numerycznie pierwiastek x^* równań $f(x) = 0$ i $g(x) = 0$ dla
   (a) $f(x) = \sin(x) - 0.37$,
   (b) $g(x) = f(x)^2 = (\sin(x) - 0.37)^2$,

   na przedziale $x \in \langle 0, \frac{\pi}{2}\rangle$ z dokładnością 10^{-6} metodami (a-d) z zad. 1 (poza przypadkami, kiedy nie da się tego zrobić). Ile kroków potrzeba, żeby osiągnąć założoną dokładność za pomocą poszczególnych metod? Zbadaj, jak zachowuje się ciąg $x_i - x^*$ dla wszystkich metod oraz funkcji f i g (dokładne rozwiązanie to oczywiście $x^* = \arcsin(0.37)$). W tym celu, zależność $x_i - x^*$ przedstaw na wykresie (należy dobrać odpowiednią skalę osi, tak, żeby wykres był czytelny). Usprawnij rozwiązanie dla funkcji $g(x)$ stosując metodę z zad. 5.

7. (*dla ambitnych*) Po opublikowaniu kodu źródłowego gry komputerowej Quake III Arena, dosyć szeroko dyskutowana była oryginalna implementacja funkcji `Q_rsqrt` z pliku `q_math.c` (zob. tutaj), obliczająca przybliżoną wartość funkcji $1/\sqrt{x}$. Jej kod, po usunięciu *zbędnych* komentarzy, to:

```c
float Q_rsqrt( float number )
{
        long i;
        float x2, y;
        const float threehalfs = 1.5F;

        x2 = number * 0.5F;
        y  = number;
        i  = * ( long * ) &y;                   // evil floating point bit level hacking
        i  = 0x5f3759df - ( i >> 1 );
        y  = * ( float * ) &i;
        y  = y * ( threehalfs - ( x2 * y * y ) );   // 1st iteration
//      y  = y * ( threehalfs - ( x2 * y * y ) );   // 2nd iteration, this can be removed
        return y;
}
```

   Spróbuj wyjaśnić jak działa funkcja `Q_rsqrt` i jaki jest jej związek z listą 7.