Paweł Troka, nr indeksu 132334

Jakub Mamelski, nr indeksu 137339

Jan Pączkowski, nr indeksu 137366

# MODELOWANIE I SYMULACJA SYSTEMÓW

Metoda Rungego-Kutty I-rzędu

ZADANIE 4: PRZYBLIŻONE ROZWIĄZYWANIE RÓWNAŃ RÓŻNICZKOWYCH

# Nazwa wybranej metody przybliżonego rozwiązywania równań różniczkowych

Wybrana metoda nazywa się „metoda Rungego-Kutty I-rzędu”, jest to de facto metoda Eulera. Człon I-rzędu w nazwie metody wywodzi się z faktu, że zależność błędu globalnego metody jest O(h1), gdzie h to krok metody.

# Warunki, które muszą być spełnione, by zadziałała

Aby metoda działała poprawnie muszą być spełnione warunki następujące:

* Warunek początkowy, czyli musimy mieć wartość x(t0)
* Warunek stabilności, czyli odpowiednio określony przyrost czasu w kroku iteracji h, szczegóły poniżej:

Odpowiednio określony krok h gwarantuje stabilność metody, okazuje się bowiem, że metoda Rungego-Kutty I-rzędu może być niestabilna zwłaszcza dla równań sztywnych[9][4], co więcej, fakt, że jest to metoda I-rzędu pogłębia jej niedokładność – jest ona wolno zbieżna. Wymaga, więc odpowiednio małego kroku iteracji h, co oznacza więcej iteracji metody, a tym samym większy czas wykonania.

Przykładowo dla liniowych równań różniczkowych postaci

y' = k y

Rozwiązanie jest niestabilne, jeżeli produkt hk=Z jest poza regionem

 \{ z \in \mathbf{C} \mid |z+1| \le 1 \}, 

Zwanym regionem liniowej niestabilności. Aby metoda działała prawidłowo, nie możemy znajdować się w obszarze jakiejkolwiek niestabilności.

Z drugiej strony h nie może być zbyt małe, bardzo mała wartość ze względu na reprezentację liczb zmiennoprzecinkowych w pamięci komputera, może po dodaniu nie zmieniać wyniku lub wprowadzać narastający błąd w innych działaniach.

Dodatkowo istnieją warunki, które muszą być spełnione, jeżeli mamy być w stanie określić górne oszacowanie błędu globalnego metody.

* x(t) powinno mieć drugą pochodną i max|x’’(t)|, w przedziale [t0,t] powinno być skończoną liczbą
* funkcja f powinna spełniać warunek Lipschitz’a, w przedziale [t0,t], dla argumentu x, tzn. musi istnieć skończona liczba L taka, że:

Gdzie ta i tb należą do przedziału [t0,tN].

# Wzory stosowane przez tę metodę w trakcie obliczeń

Na wejściu mamy postać funkcji f(x(t),t), wartość x(t0) oraz czas początkowy i końcowy (t0 i tN).

Przed rozpoczęciem obliczeń, wyznaczamy taki przyrost czasu w iteracji h, że błąd globalny metody zgodnie ze wzorem z punktu 4, jest mniejszy lub równy parametrowi ε, oznaczającemu dokładność. Aby to uzyskać przekształcamy wzór podany w punkcie 4.

Po wyznaczeniu ze wzoru z punktu 4 odpowiedniego h (spełniającego warunek, że górne oszacowanie błędu metody jest mniejsze lub równe ε (dokładności)), możemy łatwo wyznaczyć liczbę kroków metody.

Dzielimy po prostu przedział czasowy przez wyznaczoną wcześniej długość kroku iteracji.

Proces obliczania kolejnych wartości poszukiwanej funkcji x(t) jest dosyć prosty. Z warunku początkowego mamy x(0).

I teraz, w każdej kolejnej iteracji obliczamy najpierw współczynnik k1, jako iloczyn kroku i funkcji w punkcie n. Z uwagi na to, że metoda jest pierwszego rzędu, w każdej iteracji obliczamy współczynnik k1 i innych współczynników nie potrzebujemy.

Następnie obliczamy kolejną wartość czasu poprzez dodanie kroku iteracji h. To dla tej wartości czasu będzie nowo obliczona wartość funkcji x(t).

Na koniec danej iteracji obliczamy wartość kolejnego x-a, jako sumę poprzedniej wartości i wyznaczonego w tym kroku parametru k1.

Po zakończeniu całego procesu mamy stablicowaną w dziedzinie czasu od t0 do tN funkcję x(t) z obliczonym na podstawie wymaganej dokładności ε krokiem czasowym h.

# Wzór szacujący z góry błąd metody

Dosyć łatwo jest określić górną granicę dla tzw. błędu lokalnego, czyli popełnianego w jednym kroku metody [4], skąd później można wyprowadzić górne ograniczenie dla błędu globalnego metody (czyli maksymalną wartość błędu na całym przedziale wyznaczania x(t)), jak dokonano choćby w [8]. Wzór ten wygląda następująco.

 |\text{GTE}| \le \frac{hM}{2L}(e^{L(t-t_0)}-1) \qquad \qquad 

Gdzie h jest przyrostem t w kroku iteracji (tj. ti = ti-1 + h), M jest górnym ograniczeniem drugiej pochodnej x(t) na rozważanym przedziale [t0,t], L jest stałą Lipschitz’a dla funkcji f w przedziale [t0,t], t to czas końcowy, natomiast t0, to czas początkowy.

Wzór ten nie jest jednak zbyt użyteczny, ponieważ nie da się za bardzo wyznaczyć górnego ograniczenia dla x’’(t) nie znając x(t). Dlatego w programie skorzystano ze związku błędu globalnego z lokalnym i innej postaci błędu lokalnego [8][4][6], co pozwoliło zamienić M we wzorze na przedstawioną poniżej postać x’’(t0) (poniżej y=x).

y''(t_0) = {\partial f\over\partial t}(t_0, y(t_0)) + {\partial f\over\partial y}(t_0, y(t_0)) \, f(t_0, y(t_0)).

# Bibliografia

1. <http://www.kaims.pl/~robert/MISS/i03.pdf>
2. <https://pl.wikipedia.org/wiki/Algorytm_Rungego-Kutty>
3. <https://en.wikipedia.org/wiki/Runge–Kutta_methods>
4. <https://en.wikipedia.org/wiki/Euler_method>
5. <https://pl.wikipedia.org/wiki/Metoda_Eulera>
6. <http://www.cfm.brown.edu/people/sg/AM35odes.pdf>
7. <https://en.wikipedia.org/wiki/Lipschitz_continuity>
8. <http://www.maths.nuigalway.ie/~niall/MA385/2-3-ErrorAnalysis.pdf>
9. https://en.wikipedia.org/wiki/Stiff\_equation