Obliczanie wartości pochodnej funkcji na podstawie jej wartości w skończonej liczbie punktów

Aleksander Czeszejko-Sochacki

15 grudnia 2017

1 Wstęp

Problem wyliczania pochodnej funkcji na podstawie jej wartości w skończonej liczbie punktów jest dość popularny, szczególnie w zagadnieniach fizycznych. Powiedzmy, że mamy bardzo wiele obserwacji i przypuszczamy, że układają się one w pewną zależność. Chcemy obliczyć, jak ta zależność się zmienia (np. w czasie). Jest to analogiczny problem do tego, który będę rozważał w tym sprawozdaniu.

2 Cel wykonanych doświadczeń

Wpierw chcemy ustalić, czy dla każdej funkcji da się przybliżyć pochodną w danym punkcie na podstawie wartośći w węzłach. Sprawdzimy, jak dokładność przybliżenia zależy od rozmieszczenia i liczby węzłów. Ocenimy poprawność i dokładność testowanych metod.

3 Opis użytych metod

Do obliczania przybliżonej wartości pochodnej użyłem dwóch metod (zwanych w dalszej części odpowiednio metodą 1. i metodą 2.):

- Przybliżania pochodnej w punkcie jako ilorazu różnicowego na przedziale, do którego ten punkt należy
- Obliczanie pochodnej wielomianu interpolacyjnego Newtona w punkcie przechodzącego przez dane węzły

Zakładamy dalej, że $X = [x_1, x_2, \ldots, x_n]$ - zbiór wezłów, przy czym $x_1 < x_2 < \ldots < x_n$ oraz a -punkt, w którym chcemy znać przybliżenie pochodnej.

3.1 Metoda obliczania ilorazu różnicowego na przedziale

Schemat metody

1. Znajdź $i \in \{1, 2, \ldots, n\}$ takie, że $a \in < x_i, x_{i+1} >$

2. f'(a) = $\frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$

Krok 1. został zrealizowany w pliku `program.ipynb` funkcją `search_bounds(parametry)`.
Krok 2. został zrealizowany w pliku `program.ipynb` funkcją `approx_derivate(parametry)`.

3.2 Metoda obliczania pochodnej wielomianu interpolacyjnego Newtona

Schemat metody

1. Wylicz tablicę B kolejnych ilorazów różnicowych:

$$B = [f[x_1], f[x_1, x_2], f[x_1, x_2, x_3], \ldots, f[x_1, x_2, \ldots, x_n]]$$

2. Oblicz na podstawie X i ilorazów różnicowych wielomian interpolacyjny Newtona w postaci potęgowej:
   (a) Niech newton(x) = B[0]
   (b) Oblicz kolejny wielomian węzłowy p(x) w postaci potęgowej na podstawie poprzedniego (pierwszy jest tożsamościowo równy 1)
   (c) Dodaj do wielomianu newton(x) wielomian p(x) przemnożony przez kolejny iloraz różnicowy z tablicy B
   (d) Powtarzaj kroki b) - c) dopóki nie przeiterujesz całej tablicy B
   (e) Zwróć wynik: newton(x)
3. Oblicz pochodną wielomianu newton(x), f'(a) = newton'(a)

Krok 1. został zrealizowany w pliku `program.ipynb` funkcją `coefs(parametry)`
Krok 2. został zrealizowany w pliku `program.ipynb` funkcją `newton_interpolation(parametry)`
Krok 3. został zrealizowany w pliku `program.ipynb` funkcją `newton_derivate(parametry)`

Przypuszczamy, że druga metoda będzie dawała rezultaty bliższe dokładnej wartości pochodnej niż pierwsza.

4 Testy

W pliku `program_ipynb` wykonałem szereg testów powyższych metod dla funkcji: `sin(x), ln(x), exp(x)` oraz `2(x + 1)(x - 3)`. Obliczenia przeprowadziłem w arytmetyce BigFloat w języku julia, aby uniknąć niedokładności wyników, które mogłyby być spowodowane zbyt dużą precyzją arytmetyki. Testy zostały przeprowadzone dla różnych zbiorów węzłów w przedziale [0, 10] (dla funkcji `exp(x)` również w punkcie spoza niego: 30).

4.1 `sin(x)`

Po pierwsze `sin(x)` w przedziale [0, 10] jest funkcją gładką, przyjmującą skończone, na dodatek niewielkie wartości (co do modułu). To znaczy, że nie powinno być problemu z dokładną interpolacją tej funkcji na tym przedziale. Testy to potwierdzają.

1. W kazdym teście wraz ze zwiększeniem liczby węzłów odnotowujemy zmniejszenie błędu względnego (przybliżonej wartości do wartości dokładnej).

2. Metoda 2. dawała lepsze wyniki niemal we wszystkich przypadkach (wyjątek - 11 węzłów)

3. Co więcej, w każdym teście dla 101 węzłów błąd względny wynosi 0.0 (w precyzji arytmetyki BigFloat)

4.2 `exp(x)`

Funkcja `exp(x)` jest funkcją gładką. Jednak bardzo szybko rośnie. Sprawdzimy, jak przekłada się to na dokładność omawianych metod.

1. W testach dla punktów, w których funkcja `exp(x)` przyjmuje małe wartości (1.2, π) wraz ze wzrostem liczby węzłów metoda 2. zwracała wyniki obarczone mniejszym błędem względnym.

2. W obu wyżej wymienionych punktach dla 6 węzłów metoda 1. dawała zdecydowanie lepsze wyniki, dla 11 oraz 101 węzłów - metoda 2.

3. Ciekawa obserwacja - w obu wyżej wymienionych punktach metoda 1. była najmniej dokładna nie dla 6, a 11 węzłów. Co więcej, w punkcie 1.2 najdokładniejsza była dla 6 węzłów.

4. W punkcie 30 dla 100 węzłów równoodległych w przedziale [29, 31] metoda 2 zwraca zupełnie niedokładny wynik rzędu 10^{197}, podczas gdy wartość dokładna pochodnej w tym punkcie jest rzędu 10^{13}. Metoda 1 sprawdza się w tym punkcie dość dobrze (aczkolwiek powinniśmy zdefiniować, jakiej precyzji żądamy) - błąd względny jest rzędu 10^{-5}.

4.3 `2(x + 1)(x - 3)`

Wydaje się, że interpolacja wielomianu drugiego stopnia wielomianami 6, 11 czy 101 stopnia powinna generować na tyle małe błędy przy obliczaniu pochodnej, że będą one mniejsze niż precyzja arytmetyki BigFloat. Okazuje się, że tak nie jest.

1. W punkcie 1 (w którym pochodna powyższego wielomianu się zeruje) metoda 2 dla 101 węzłów zwraca błąd względny rzędu 10^{-61} (jest on bardzo mały, ale niezerowy). Inaczej ma się w przypadku 11 i 101 węzłów oraz dla metody 1 i 6 węzłów.

2. Jest to jedyna z testowanych funkcji, dla której obie metody już dla 6 węzłów obliczyły dokładną wartość pochodnej (w arytmetyce BigFloat).

3. W punktach 5.123 i 9.9 metoda 2. okazała się niezawodna dla wszystkich zbiorów węzłów.

4.4 `ln(x)`

Czy asymptota w przedziale, w którym przybliżamy pochodną wpływa na dokładność przybliżeń? Sprawdźmy.

1. Zauważmy, że testowanie dla zbiorów węzłów zawierających zero nie ma sensu dla metody 2, ponieważ przy wyliczaniu ilorazów różnicowych bierzemy $f(0) = \infty$. Potwierdza to nasz pierwszy test dla wartości 0.1.

2. Testowanie dla zbioru węzłów takiego, że punkt, w którym przybliżamy pochodną leży w przedziale $< x_0, x_1 >$ nie ma sensu dla metody 1. Również potwierdza to nasz pierwszy test.

3. Metoda 2. w punkcie 0.1 nawet jeśli zwraca wynik, to zupełnie niepoprawny

4. Metoda 1. w punkcie 0.1 dla węzłów bez 0 zwraca relatywnie dobry wynik dla 1000 węzłów.

5. W punkcie 1 metoda 2 również nie zwraca satysfakcjonujących wyników, jednak błędy są dużo mniejsze niż w punkcie 0.1.

5 Wnioski

Jak mogliśmy się przekonać w powyższych testach, nie dla każdej funkcji opisane przeze mnie metody dawały satysfakcjonujące rezultaty (`ln(x)`). Przy obliczaniu przybliżonej wartości pochodnej w punkcie z danego przedziału pomocne mogą okazać się informacje na temat tej funkcji:

- Czy funkcja w tym przedziale nie ma asymptoty/asymptot?
- Czy funkcja w tym przedziale jest określona w każym punkcie tego przedziału?

Dla funkcji, które spełniają oba powyższe kryteria możemy wyciągnąć następujące wnioski:

- Nie zawsze zwiększenie liczby węzłów powoduje zwiększenie precyzji przybliżenia
- W zdecydowanej większości metoda 2 dawała lepsze rezultaty niż metoda 1. Ważny wyjątek: metoda 1. zwracała lepszy wynik na przedziale, na którym funkcja bardzo szybko rośnie (`exp(x)`). Dlaczego w tym przypadku metoda 1. zwróciła relatywnie dobry wynik? Otóż gdy funkcja na pewnym przedziale bardzo szybko rośnie, bardziej przypomina na pewnym małym przedziale między węzłami funkcję liniową niż w przedziale, w którym funkcja zachowuje się inaczej (np. funkcja `exp(x)` w przedziale [0, 2] nie przypomina funkcji liniowej - jest "zaokrąglona". A metoda 1. bazuje właśnie na założeniu, że przybliżana funkcja jest na pewnych małych przedziałach liniowa.
- Metoda 2 świetnie sprawdzała się dla wielomianu `2(x + 1)(x - 3)`. Ważne - wielomian był niższego stopnia niż wielomiany interpolacyjne Newtona z metody 2.