Metody numeryczne w elektrotechnice

Ćwiczenie 6 — Różniczkowanie i całkowanie numeryczne

Dariusz Borkowski

18 kwietnia 2012

Wstęp

Na zajęciach rozwiążecie 4 zadania. Pierwsze pokaże wam zastosowania i możliwe problemy związane z numerycznym różniczkowaniem. Drugie dotyczy całkowania funkcji przybliżającej dane pomiarowe znane w dyskretnych punktach (w dwóch wymiarach). Trzecie i czwarte to w zasadzie to samo zadanie tylko realizowane raz w Matlabie, raz w C. Jego przedmiotem jest wyznaczenie całki oznaczonej z funkcji, której postać analityczna jest znana.

1 Zastosowanie różniczkowania do analizy skoków narciarskich – 1 pkt

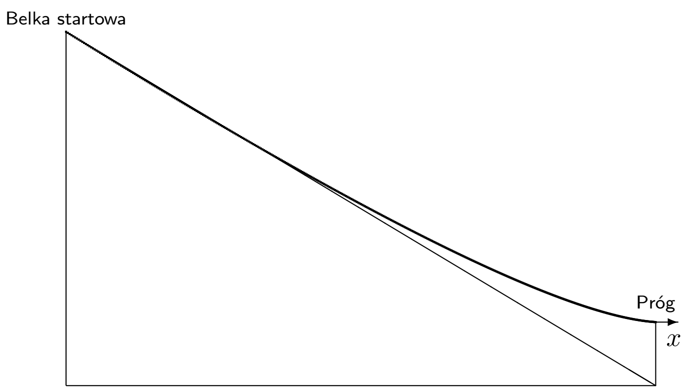
Jednym z najważniejszych parametrów wpływających na długość skoków narciarskich jest prędkość skoczka na progu, której średnia dla tej skoczni wynosi $92\ \mathrm{km/h}$ czyli $25,55\ \mathrm{m/s}$. Zwiększenie tej prędkości o $1\ \mathrm{km/h}$ skutkuje wydłużeniem skoku o około $10\ \mathrm{m}$. Z powodu spadku formy jednego ze skoczków narciarskich dostałeś zadanie wykonania analizy prędkości i przyspieszenia skoczka w pierwszych fazach skoku, t.j. fazie najazdu i fazie odbicia.

Pomiary wykonałeś na Wielkiej Krokwii w Zakopanem. Rzeczywiste parametry tej skoczni są następujace: wysokość $141\ \mathrm{m}$, długość rozbiegu $92\ \mathrm{m}$, nachylenie progu $10,5^0$ (rysunek 1). Pomiary polegały na pomiarze czasu, w którym skoczek przejechał przez fotodetektory rozmieszczone co 4 metry wzdłuż rozbiegu skoczni. Czas (w sekundach) był mierzony z dokładnością do czwartego miejsca po przecinku. Z pomiarów otrzymałeś 2 wektory: wektor czasu $\boldsymbol{t} = \begin{bmatrix} t_1 & t_2 & \dots & t_N \end{bmatrix}^T$ oraz wektor położeń $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_N \end{bmatrix}^T$. Aby wyznaczyć na podstawie tych danych prędkość i przyspieszenie skoczka musisz policzyć przybliżone wartości pierwszej i drugiej pochodnej po czasie w punktach pomiaru.

Przybliżenie pierwszej pochodnej położenia $x(t)$ w punkcie t_i możesz obliczyć np. jako wsteczny iloraz różnicowy:

$$\left.\frac{dx(t)}{dt}\right|_{t=t_i} \approx \frac{\Delta x_i}{\Delta t_i} = \frac{x_i - x_{i-1}}{t_i - t_{i-1}} \tag{1}$$

Możesz korzystać również z ilorazów wprzód oraz centralnych (te ostatnie dają lepsze oszacowanie pochodnej), ale w zależności od wybranego sposobu przybliżania pochodnej, nie otrzymasz przybliżenia pochodnej w punkcie x_1 lub x_N lub obydwu. W tym przypadku najważniejsza była prędkość w ostatnim punkcie pomiarowym. Pamiętaj, że punkty czasowe, w których dokonano pomiaru nie są równoodległe, równoodległe są tylko położenia fotodetektorów, co powoduje, że nie możesz przyjąć stałego mianownika $\Delta t_i = h = \mathrm{const}$



Rysunek 1: Schematyczny rysunek rozbiegu skoczni Wielka Krokiew w Zakopanem.

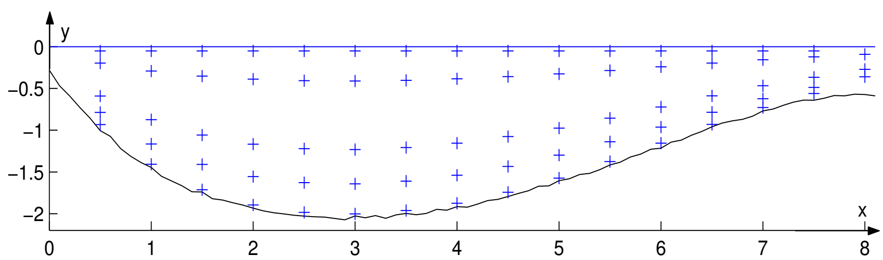
we wzorze (1). Przybliżenie drugiej pochodnej położenia x możesz uzyskać po wyprowadzeniu analogicznego wzoru na drugą pochodną lub poprzez różniczkowanie obliczonej wcześniej pierwszej pochodnej. Jak wiesz z wykładu każda operacja różniczkowania numerycznego wzmacnia błędy, co powininno być już widać w przybliżeniu drugiej pochodnej.

Otwórz plik `lab06z01.m` zawierający dane pomiarowe. Wyznacz i narysuj na osobnych wykresach przebiegi położenia, prędkości i przyspieszenia skoczka w funkcji czasu. Określ prędkość skoczka na progu, spróbuj wyjaśnić dlaczego zmniejsza się jego przyspieszenie oraz dlaczego prędkość przed progiem mogła zmaleć (w analizie pomocny byłby zapis z kamery poruszającej się wzdłuż rozbiegu razem ze skoczkiem). Zmniejsz precyzję danych pomiarowych (wektor czasu) przez zaokrąglenie i usunięcie czwartego miejsca po przecinku, a następnie narysuj znów przebiegi prędkości i przyspieszenia. Co jest istotne przy pomiarach sygnałów które mają być różniczkowane?

Zastanów się czy do niniejszego zadania można zastosować cyfrowe filtry różniczkujące. Uzasadnij odpowiedź.

2 Obliczanie przepływu średniego przez całkowanie w dwóch wymiarach – 3 pkt

Otrzymałeś zadanie określenia średniego przepływu miesięcznego pewnej rzeki o nieregularnym korycie. Rzeka ta ma około 10 m szerokości w przekroju pomiarowym i największą głębokość w przekroju około 2 m. Zgodnie z zasadami pomiarów hydomerycznych przez miesiąc wykonano punktowe pomiary prędkości przepływającej wody w szesnastu pionach hydrometrycznych co $0,5$ metra. W każdym pionie wykonano 3 lub 5 pomiarów w zależności od głębokości y_m w tym pionie. Dla głębokości $y_m \leqslant 0,6$ m pomiary wykonano na wysokościach $y = \{-0,2y_m; -0,6y_m; -0,8y_m\}$ licząc od zwierciadła wody. W przekrojach o głębokości $y_m > 0,6$ m wykonano dodatkowe pomiary 5 cm nad dnem i 5 cm poniżej zwierciadła wody. Po zakończeniu pomiarów prędkości zmierzone w ciągu całego miesiąca w każdym punkcie uśredniono. Prędkość wody była mierzona nowoczesnym miernikiem ultradźwiękowym, w którym nadajnik i odbiornik umieszczone były w jednej linii równolegle do linii prądu. Pomiarów w pobliżu prawego brzegu nie wykonano ze względu na jego zarastanie, można tam przyjąć zerową prędkość. Schematyczny rysunek przekroju koryta wraz z punktami pomiarowymi pokazano na rysunku 2.



Rysunek 2: Kształt dna w przekroju pomiarowym, zwroty osi i położenia punktów pomiarowych.

Poszukiwany przepływ określony jest wzorem

$$Q = v_{sr} F \qquad \left[\frac{\mathrm{m}^3}{\mathrm{s}}\right] \tag{2}$$

gdzie F jest powierzchnią przekroju pomiarowego, a v_{sr} średnią prędkością wody w tym przekroju. Prędkość ta określona jest wzorem

$$v_{sr} = \frac{1}{x_m} \int\limits_0^{x_m} \left(\frac{1}{y_m(x)} \int\limits_{-y_m}^{0} v(x,y)dy \right) dx \qquad \left[\frac{\mathrm{m}}{\mathrm{s}}\right] \tag{3}$$

Najprostsze byłoby policzenie średniej ze wszystkich punktowych prędkości wody, jednak przy tak rzadko rozłożonych punktach pomiarowych wynik byłby obarczony sporym błędem. Chcąc otrzymać dokładniejszy (dokładny nigdy nie będzie znany) wynik i wiedząc, że prędkości wody w funkcji położenia w pionie i poziomie powinny się zmieniać w sposób ciągły (a nie skokowy) można poprowadzić funkcję interpolującą wartości prędkości (w dwóch wymiarach) i funkcję tą całkować według wzoru 3. Łatwiej można rozwiązać to zadanie przez wyznaczenie funkcji interpolacjących prędkość w poszczególnych pionach hydrometrycznych, obliczenie ich całek (po dy), a następnie dokonanie kolejnej interpolacji uzyskanych wyników i policzenie drugiej całki (po dx). Do przybliżenia całki z funkcji $f(x)$ jednej zmiennej można wykorzystać kwadratury złożone różnych rzędów. Wzory przybliżające całkę z $f(x)$ na przedziale od a do b dla węzłów równoodległych ze skokiem $h = \frac{b-a}{m}$ pokazano poniżej. Mogą to być np. metoda prostokątów

$$\int_a^b f(x)dx \approx \underbrace{h \sum_{k=0}^{m-1} f_k}_{wprzod} \approx \underbrace{h \sum_{k=1}^{m} f_k}_{wstecz} \tag{4}$$

metoda trapezów

$$\int_a^b f(x)dx \approx \frac{h}{2} \sum_{k=0}^{m-1} (f_k + f_{k+1}) \tag{5}$$

lub metoda Simpsona

$$\int_a^b f(x)dx \approx \frac{h}{3} \sum_{k=1}^{m/2} (f_{2k-2} + 4f_{2k-1} + f_{2k}) \tag{6}$$

Na podstawie danych pomiarowych zawartych w pliku `lab06z02.mat` wyznacz średni miesięczny przepływ Q $[m^3/s]$ w przekroju. W tym celu oblicz średnią prędkość v_{sr} w każdym pionie hydrometrycznym przybliżając wartości całki w pionie i dzieląc je przez szerokość przedziału całkowania (zazwyczaj głebokość w pionie). Nastepnie scałkuj uzyskane wartości średniej prędkości w pionach i znów podziel je przez szerokość przedziału całkowania (zazwyczaj szerokości koryta rzeki). Tak wyznaczoną prędkość średnią pomnóż przez pole powierzchni przekroju $F = 11.3846\ \mathrm{m}^2$, aby uzyskać przepływ Q.

Przybliżone wartości całek w poszczególnych pionach hydrometrycznych wyznacz metodami prostokątów i trapezów na węzłach równoodległych. Ponieważ punkty pomiaru w pionach nie są równoodległe i jest ich mało, musisz dokonać interpolacji prędkości w poszczególnych pionach. Przeprowadź interpolację wielomianową danych węzłów pomiarowych poleceniem `polyfit`[1]. Następnie wyznaczone funkcje interpolujące prędkości w pionach spróbkuj poleceniem `polyval` ze stałym krokiem $5\ \mathrm{cm}$ licząc od od lustra wody aż do dna t.j. w Matlabie dla wysokości `0:-0.05:-ym`. Do przechowania wyników (czyli wartości y i v w każdym pionie) wykorzystaj tablicę typu `cell array` zawierającą struktury, z których każda przechowuje wektory `y` i `v` w danym pionie. Ostatecznie, wyznaczone wartości średnich prędkości w pionach należy scałkować (tym razem w poziomie) i podzielić przez szerokość przedziału całkowania. Całkowanie wykonaj metodami prostokątów i trapezów bez interpolacji, gdyż tym razem węzły całkowania (piony) są równoodległe, rozmieszczone co $0,5\ \mathrm{m}$. Szerokość przedziału całkowania wynosi 8 m.

Porównaj wyniki uzyskane przez policzenie średniej prędkości funkcją `mean` oraz metodami prostokątów i trapezów. Wykonaj to samo zadanie wyznaczając węzły całkowania (punkty próbkowania funkcji interpoluących w pionach) co $20\ \mathrm{cm}$. Porównaj wyniki z uzyskanymi poprzednio.

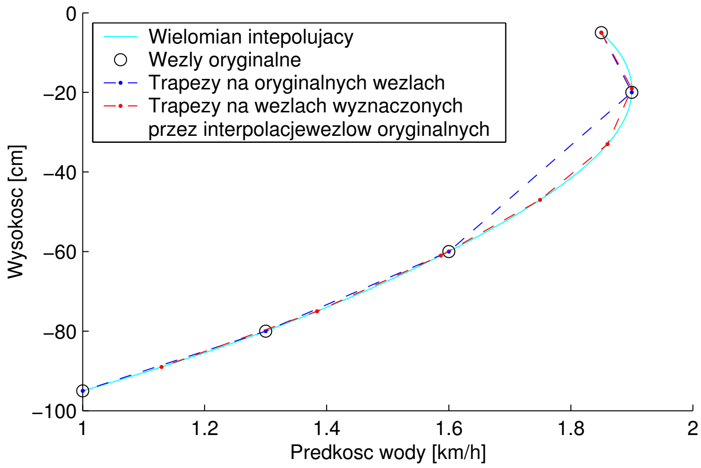
Pamiętaj, że przybliżając wartość całki kwadraturą dowolnego rzędu i tak wykonujesz interpolację węzłów wielomianem (np. stopnia zerowego dla metody prostokątów, pierwszego dla trapezów, drugiego dla Simpsona). Interpolacja krzywymi sklejanymi (lub wielomianem) w pionach hydrometrycznych była potrzebna dla zwiększenia dokładności przybliżenia całki obliczanej metodą prostokątów lub trapezów (rys. 3). Użycie kwadratur wyższych rzędów nawet na pierwotnych węzłach mogłoby prowadzić do znacznie dokładniejszych przybliżeń całki i nie byłaby potrzebna dodatkowa interpolacja. Przy regularnym kształcie całkowanej funkcji może być to korzystniejsze, gdyż zbliżoną dokładność uzyskać można w krótszym czasie przy dłuższym kroku całkowania, czyli mniejszej ilości obliczeń.

Prędkości wody w przekroju pomiarowym pokazano na rysunku 4 . Na rysunku 5 pokazano tachoidy[2] uzyskane przez interpolację danych pomiarowych w pionach.

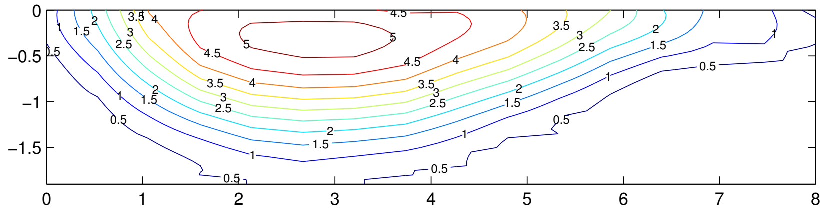
Plik `lab06z02.m` zawiera szkielet programu. Plik `lab06z02.mat` zawiera dane pomiarowe. Dane zapisano w tablicy `vpom` typu `cell array` zawierającej 16 struktur opisujących pomiary w każdym pionie

[1] Zastosowanie funkcji interpolującej w postaci wielomianu zamiast funkcji sklejanej ma związek, z faktem iż pomiary wykonane zostały nie od lustra wody do dna lecz w mniejszym zakresie, a całkować mamy po całej głębokości, a zatem poza obszarem pomiaru mamy do czynienia z ekstrapolacją. W tym wypadku lepszy będzie wielomian niż funkcja sklejana.

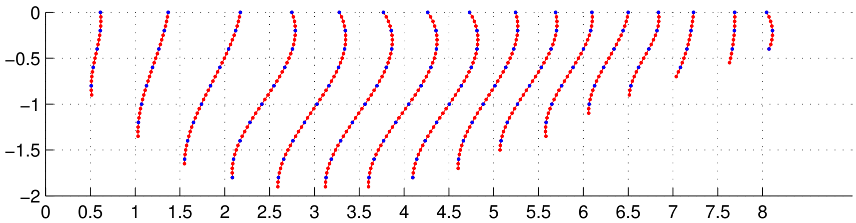
[2] Tachoidy to rozkłady prędkości wody w kolejnych pionach hydrometrycznych. Oddalenie punktu tachoidy od pionu hydrometrycznego (w prawą stronę) jest proporcjonalne do prędkości wody w danym punkcie.



Rysunek 3: Porównanie całkowania metodą trapezów na różnych węzłach.



Rysunek 4: Prędkości wody w przekroju pomiarowym [km/h].



Rysunek 5: Tachoidy w kolejnych pionach hydrometrycznych.

hydrometrycznym. Każda struktura składa się z dwóch wektorów jednakowej długości. Wektor `y` zawiera współrzędne pionowe y na, których dokonano pomiaru, a wektor `v` zawiera wartości prędkości wody v w tych punktach. Wartości prędkości podano w km/h (pamiętaj o przeliczeniu jednostek!). Tablica typu `cell array` jest korzystna, ponieważ może zawierać różne typy danych o różnym rozmiarze (np. struktury).

A teraz przykład korzystania ze struktur i tablic typu `cell array`. Do drugiego elementu tablicy `ca` typu `cell array` odwołujemy się przez `ca{2}`. Do wektora (elementu) `vec` struktury `str` odwołujemy się jak w języku C czyli `str.vec`. Do pierwszego elementu wektora `vec` odwołujemy się przez `vec(1)`. Po zebraniu tych informacji razem do pierwszego elementu wektora `vec` będącego składnikiem struktury będącej drugim elementem tablicy `ca` typu `cell array` odwołujemy się przez `ca{2}.vec(1)`. Utworzyć nową strukturę `str` zawierającą wektory `[1 2 3 4]` i `[2 1]` można przez `str.a = [1 2 3 4]` i `str.b = [2 1]`. Utworzyć nowy obiekt `ca` typu `cell array` zawierający jako drugi element strukturę `str` można przez `ca{2} = str`. Przećwicz to sobie. Prawda że proste?

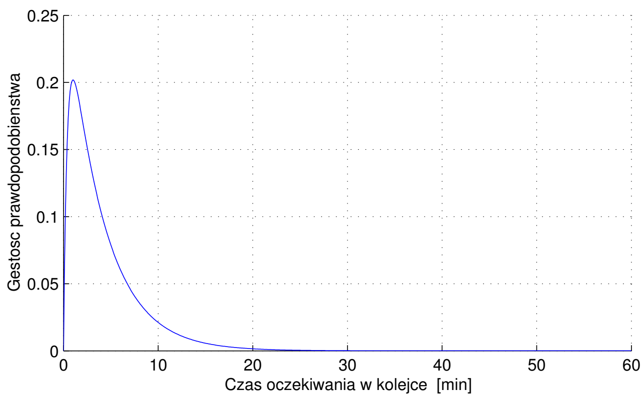
Zadanie dla ambitnych za dodatkowy punkt! Do wyznaczenia prędkości średniej na podstawie już policzonych prędkości średnich w pionach użyj adaptacyjnej metody Simpsona zaimplementowanej w funkcji `quad`. Funkcja `I = quad(fun,a,b)` wykorzystuje definiowaną przez użytkownika funkcję `fun` (najczęściej będącą w osobnym m-pliku), której argumentem jest wektor x określający węzły całkowania, a która zwraca wektor wartości całkowanej funkcji $f(x)$ w tych węzłach. W tym wypadku funkcja `fun` ma zwracać wartości funkcji interpolującej średnie wartości prędkości w pionach.

3 Wyznaczanie prawdopodobieństwa zdarzeń przez całkowanie rozkładu – 1 pkt

Zapewne nieraz czekałeś w kolejce do dziekanatu, który przyjmuje do godziny 13:00 i zastanawiałeś się czy zdążysz się dostać do środka przed zamknięciem. Załóżmy, że jest przyszedłeś o godzinie 12:50, odebrałeś bileciek z Twoim numerem w kolejce, a na z informacji na wyswietlaczu dowiedziałeś się, że średni czas oczekiwania na wejście 3 minuty, a przed Tobą w kolejce stoi 20 osób. Z prostego rachunku wynika, że przewidywana obsługa oczekujących przed Tobą studentów zajemie 1 godzinę. Mógłbyś stwierdzić, że nie warto czekać, ale równie dobrze, może się okazać, że sporo studentów zostanie obsłużonych szybciej bądź odesłanych „z kwitkiem" z powodu braku potrzebnych dokumentów. Przeanalizować ten problem możesz w następujący sposób sposób. Jeśli prawdopodobieństwo zdarzenia, że zdążysz wejść przed zamknięciem jest większe od $1/2$ to czekasz w kolejce, a jeśli jest mniejsze to idziesz z kolegami do knajpy. Żeby wyznaczyć prawdopodobieństwo musisz znać funkcję gęstości prawdopodobieństwa opisującą tą sytuację. W dużym uproszczeniu sytuację tą opisuje rozkład gęstości prawdopodobieństwa dany wzorem

$$f(t_o) = \frac{1}{t_1 - t_2}\left(e^{-\frac{t_o}{t_1}} - e^{-\frac{t_o}{t_2}}\right) \tag{7}$$

t_o jest czasem oczekiwania w kolejce, t_1 i t_2 są parametrami rozkładu. Jako wartości parametrów przyjmij[3] $t_1 = 0,4\ \mathrm{min}$, $t_2 = 3,8\ \mathrm{min}$. Jednostka czasu to jedna minuta. Rozkład ten pokazano na rysunku 6.



Rysunek 6: Rozkład gęstości czasu oczekiwania w kolejce.

Poszukiwane prawdopodobieństwo zdarzenia, że czas oczekiwania w kolejce t_o będzie krótszy niż 10 minut, które pozostały do zamknięcia dziekanatu, wyraża się całką oznaczoną z funkcji gęstości prawdopodobieństwa

$$P(t_o < 10) = \int_0^{10} f(t)dt \tag{8}$$

Wyznacz w Matlabie prawdopodobieństwo, że dostaniesz się do dziekanatu przed zamknięciem całkując rozkład dany wzorem (7) adaptacyjną metodą Simpsona (funkcja quad) oraz metodą prostokątów (zaaplikowaną samodzielnie), dla węzłów całkowania co 1 minutę. Oceń czy warto czekać w kolejce. Porównaj wartości otrzymane dwiema metodami. Wyjaśnij na rysunku skąd tak duże różnice?

Jak wiesz, całka od minus do plus nieskończoności z dowolnej funkcji gęstości prawdopodobieństwa jest równa 1 (określa prawdopodobieństwo zdarzenia pewnego). Sprawdź to całkując funkcję gestości 7 w przedziale od $a = 0$ do b gdzie za b przyjmiesz kolejno $100, 1000, 10000$. Całkowanie wykonaj również dwiema metodami. Porównaj wyniki ze znaną wartością całki dla zdarzenia pewnego.

4 Obliczanie całek w języku C – 3 pkt

Oblicz całkę (7) z poprzedniego zadania korzystając z trzech (spośród wielu) funkcji całkowania udostępnianych przez bibliotekę GSL. Będą to:

[3] Poznanie rzeczywistych wartości parametrów tego rozkładu jest praktycznie niemożliwe. W praktyce często spotykamy potrzbę całkowania funkcji gęstości prawdopodobieństwa np. rozkładu normalnego o znanych parametrach μ i σ.

- `gsl_integration_qng  // najprostsza (w GSLu) metoda nieadaptacyjna`
- `gsl_integration_qag  // najprostsza metoda adaptacyjna`
- `gsl_integration_qags // adaptacyjna dla funkcji z osobliwościami`

Załaduj plik `lab06z04.c` zawierający szkielet programu. Uzupełnij program i sprawdź przy jakich wartościach końca przedziału całkowania b programy przerywają obliczenia lub dają złe wyniki. Wartości te będą inne dla każdej z funkcji i są częściowo zależne od parametrów wywołania funkcji obliczających całkę.

Zapoznaj się (i postaraj zrozumieć) w jaki sposób definiowane są w GSLu funkcje przeznaczone do całkowania (obiekty typu `gsl_function`). Jest to ważne, ze względu na częstą potrzebę użycia w programie funkcji o kilku parametrach danej wzorem analitycznym. Zwróć uwagę na uniwersalny sposób przekazywania parametrów do funkcji za pomocą wskaźnika na obiekt typu `void`. W tym wypadku do funkcji oprócz zmiennej t_o przekazywana jest struktura zawierająca dwa parametry t_1 i t_2 całkowanego rozkładu.

Poniżej opisałem niezbędne w tym zadaniu funkcje biblioteki GSL służące do całkowania numerycznego (funkcji w jednym wymiarze). Funkcji tej klasy jest więcej, między innymi są takie, które potrafią obliczać całki oznaczone od $-\inf$ do $\inf$. Zestaw tych funkcji jest reimplementacją w C alogrytmów fortranowego pakietu Quadpack[4].

`int gsl_integration_qng (const gsl_function *f, double a, double b, double epsabs, double epsrel, double * result, double * abserr, size_t * neval)`. Funkcja wykorzystuje 10, 21, 43 i 87 punktowe reguły całkowania metodą Gaussa–Kronroda[5]. Przyjmuje wskaźnik `*f` na całowaną funkcję $f(x)$ typu `gsl_function`. Argumenty `a` i `b` to początek i koniec przedziału całkowania, `epsabs` i `epsrel` są limitami bezwzględnego i względnego błędu całkowania. Wynik całkowania umieszczany jest pod adresem, na który wskazuje `result`, szacowany błąd całkowania jest zapisywany pod adresem `abserr`, a pod adresem `neval` zapisywana jest ilość wyznaczeń wartości całkowanej funkcji.

`int gsl_integration_qag (const gsl_function *f, double a, double b, double epsabs, double epsrel, size_t limit, int key, gsl_integration_workspace * workspace, double * result, double * abserr)`. Ta funkcja dobiera adptacyjnie krok całkowania funkcji `f` w przedziale od `a` do `b`. Całkowanie wykorzystuje jak powyżej metodę Gaussa–Koronoda. Argumenty o identycznych nazwach takich jak wyżej mają znaczenie takie jak wyżej. Argument `key` wypływa na wybór rzędu kwadratury[6] i może przyjmować wartości od 1 do 6 (patrz dokumentacja). W każdej iteracji adaptacyjnego całkowania przedział o największej estymowanej wartości błędu jest dzielony na podprzedziały, które są zapamiętywane w pamięci zaallokowanej jako `workspace`. Argument `limit` określa maksymalną liczbę podprzedziałów, która nie może przekroczyć rozmiaru zaalokowanego przez `workspace`.

`int gsl_integration_qags (const gsl_function * f, double a, double b, double epsabs, double epsrel, size_t limit, gsl_integration_workspace * workspace, double *result, double *abserr)`. Ta funkcja wykorzystuje adaptacyjną 21–punktową metodę Gaussa–Kronroda do całkowania funkcji z nieciągłościami i całkowalnymi osobliwościami[7]. Argumenty wywołania są takie same jak w opisanej wyżej funkcji.

`gsl_integration_workspace * gsl_integration_workspace_alloc (size_t n)` allokuje pamięć (miejsce pracy) dla adaptacyjnych metod całkowanie (jak opisana powyżej) i zwraca wskaźnik do obiektu typu `gsl_integration_workspace`, w którym przechowywane będą podprzedziały, wyniki ich całkowania i estymaty błędów.

Na następne zajęcia

Zastosowania wartości własnych macierzy.

[4] http://www.netlib.org/quadpack/
[5] Zainteresowanych zasadą działania metody odsyłam do dokumentacji GSLa dotyczącej całkowania
[6] Kwadratury wyższych rzędów dają dokładniejsze wyniki dla funkcji o gładkim przebiegu, itd.
[7] Aby się dowiedzieć jak to działa znowu odsyłam do dokumentacji.