

Semestr letni 2002/2003

Konsultacje: poniedziałek 10-11, najlepiej wcześniej umówić się e-mailem, pok.1020 (na parterze koło biblioteki). Należy też sprawdzić: Plan

Spis treści:

1. Wstęp do informatyki - §1
2. Równania Różniczkowe z labem - §2

1 Wstęp od informatyki

Poniedziałek 8:30-10. Sala 3190 lub Lab. Lab przeplata się z ćwiczeniami przy tablicy - co drugi tydzień lab, ale może to się zmienić w miarę potrzeb! Tematy na lab podawane są na wyrost i być może z pewnym wyprzedzeniem - zachęcam do samodzielnego wykonywania części zadań.

Zaliczenie: projekt (program komputerowy - 50-60 reszta pktów. Zaliczenie od koła 50-55

Kartkówki: -pierwsza - 24 marca 2003 (wstępny termin) - zakres - wskaźniki, listy 1- i 2-kierunkowe, drzewa BST (ewent. AVL)- wstawianie i usuwanie elementów drzew, postać liniowa drzewa binarnego itp Czas 45-60 minut. Przykładowe zadania - na kartkówce będą podobne o ile nie takie same:

1. Napisać procedurę która dla danej listy 1-wskaźnikowej zawierającej klucze całkowite i danych k, l zwróci wskaźnik do nowej listy 1-wskaźnikowej zawierającej wszystkie elementy pierwotnej listy o tych kluczach - procedura ma usunąć te elementy z listy wyjściowej (modyfikując ją); ma zwrócić wskaźnik nil o ile takowych elementów nie ma albo jeśli lista wyjściowa pusta.
2. Wstaw do BST kolejno elementy 2,13,56,3,9,23,67,45. Rysując kolejne postaci drzewa w formie diagramu, następnie usuń 23 znowu rysując postać drzewa. Wypisz postać liniową drzewa po usunięciu 23. (Ewentualnie to samo dla AVL)
3. Dla danej postaci liniowej drzewa binarnego (np (3 (4)) choć na kartkówce na pewno dużo dłuższej) narysuj postać w postaci diagramu - drzewa. Czy jest to drzewo BST lub ewent. AVL?
4. Napisz procedurę która znajduje element najmniejszy w drzewie BST
(type element_drzewa=record parent, left, right: wsk end; wsk=^element_drzewa)

Zadania komputerowe: Można albo oddać jeden większy projekt albo zebrać pkty z kilku mniejszych, warunek zaliczenia labu (obowiązkowy to min. 21 pktów z zadań komputerowych na 40pkt - które można zdobyć max). Za działanie programu wg opisu 90 programu np czy są komentarze czy struktura programu możliwie prosta itp Część prostszych zadań będzie częściowo na labie i/lub ćwiczeniach tablicowych.

1. Zadanie z poprzedniego semestru dla ambitnych eliminacja Gaussa. (40pktów)
2. MergeSort w wersji rekurencyjnej dla l. całkowitych z porządkiem \leq z wykorzystaniem list 1-wskaźnikowych (jak na ćwiczeniach tablicowych). (14pkt). Program ma wczytać powiedzmy liczby całkowite z pliku posortować i zapisać w innym pliku. Musi istnieć możliwość obejrzenia kolejnych etapów algorytmu np sklejanie itd

3. Wstawianie i usuwanie węzłów z drzewa BST zaimplementowanego wskaźnikowo, znajdowanie odpwiedznych elementów - musi istnieć możliwość wypisania na ekran wersji liniowej drzewa w celu obejrzenia jego struktury. Dane to stringi z porządkiem leksykalnym. (Ewentualnie jako drzewa AVL - dużo trudniejsze!) (20pktów w wersji AVL 40pktów)
4. QuickSort i randomized QuickSort w wersji rekurencyjnej na tablicy dla liczb całkowitych. Testowanie jak dla MergeSort. Powinna być możliwość wyłączania losowania tzn zeby w programie mozna bylo wybierac czy chcemy sortowac Quicksortem czy randomized QS (14pkt).
5. HeapSort w wersji tablicowej - jak na wykładzie. Testowanie i opcje jak MergeSort np. wypisanie wczytanych danych nieposortowanych, potem wypisanie kopca i na koncu danych posortowanych. (12pkt)

Na labie: listy 1-2 wskaźnikowe, implementacja stosu jako listy 1-wskaźnikowej, zastosowanie do sprawdzenia czy nawiasy w danym pliku tekstowym są dobrze sparowane, funkcja sklejana 2 posortowanych list 1-wskaźnikowych w jedną posortowaną listę 1-wskaźnikową, zastosowanie do implementacji /MergeSort/ (wersja rekurencyjna), ewentualnie wersja nierekurencyjna - dla N potęgi dwa albo dla przypadku ogólnego - (2/03) BST- wypisywanie postaci liniowej (rekurencyjne lub nierekurencyjne - trudniejsze, wstawianie do BST, usuwanie z BST, znajdowanie węzła o danym kluczu. Znajdowanie najmniejszego/największego elementu w BST-(16/03) Wersja rekurencyjna /QuickSort/ i randomized QS na tablicy-(30/03)

Aktualna ilość pktów - plik pdf.

2 Równania różniczkowe zwyczajne z laboratorium

Laboratorium - czwartek 8:30-10, Lab 2(co 2 tygodnie). Ćwiczenia - czwartek 12:15-13:45, sala 3160.

Kolokwia: pierwsze 27 marca 2003 (czwartek na cwiczeniach o 12:15)

- Program laboratorium - §2.1
- Plan wykładu - link do pliku w formacie pdf¹

Literatura: Polecam pozycje: [1, Arnold] - klasyczna książka do teorii równań różniczkowych zwyczajnych, [3, Matwiejew] - książka w której są szczegółowo opisane różne metody szukania rozwiązań w sposób klasyczny (tzn nienumerycznie), [2, Matwiejew] - zbiór zadań, [4, Ombach] - Tylko teoria rrrz, bez schematów numerycznych ale za to sporo o Maple'u i ostatnia pozycja - szczególnie zgodna z programem wykładu [5, Palczewski] - Teoria równań różniczkowych i schematów, trochę o maple'u. I wiele innych, również w języku polskim.

Powrót do mojej strony domowej

2.1 Laboratorium z Równań Różniczkowych zwyczajnych (*)

wiosna 2003, czwartki godz. 8:30-10 (co drugi tydzień, pierwsze zajęcia: 20 lub 27 luty 2003 - termin mogą się zmienić w trakcie semestru np z powodu świąt).

¹ <http://www.mimuw.edu.pl/~lmarcin/lab/rrzw02-03-pw.pdf>

2.1.1 Program labu

W trakcie zajęć na labie będziemy się zapoznawać z dwoma pakietami: *maple* i *scilab* - pierwszy jest pakietem symbolicznym - choć ma też pewne algorytmy numeryczne, drugi jest pakietem numerycznym bez obliczeń symbolicznych. *Scilab* jest klonem *matlaba* - jest to oprogramowanie free-ware - można ściągnąć ze stron (Inria-scilab) i zainstalować pod systemami: linux, windows 2000 i inne. Zaletą *maple'a* jest to że jest dużo książek do RRzw w których niektóre możliwości *maple'a* są opisane.

Część zadań będzie polegała na zaprogramowaniu prostych schematów i np zbadaniu rzędu zbieżności (dokładniejsze definicje będą na wykładzie, ćwiczeniach) - najprościej będzie to zrobić pod *scilabem* gdzie i wizualizacja jest b. prosta ale jak ktoś chce to może pisać programy w dowolnym języku np *pascal*, *C*, *C++* i rysować wykresy zgranych na dysk danych *scilabem* ewent. *maplem* albo innym narzędziem np. *gnuplotem* (można i uruchomić grafikę pod *linuxem* bezpośrednio z programu napisanego np w *C++* ale to dość trudne).

Plan labu będzie czasami podany zapewne na wyrost i w miarę możliwości zadania nie zrealizowane na danym labie przesuną się na następny.

- Lab 1 - Rysowanie pól wektorowych i przybliżonych rozwiązań przy pomocy apletów java *DFIELD* i *PPPLANE*: na stronie [www²](http://www.math.rice.edu/~dfield/); wstępne zapoznanie się z systemem *scilab* i być może *maple'a* - wprowadzanie funkcji w *scilabie* inline lub ze skryptu lub przy pomocy funkcji *deff()*- rysowanie wykresów funkcji itp. Zmiana hasła na kontakach. Przykładowy skrypt *scilaba* (uruchamiany *exec intro.sci*), plik tekstowy z krótkim opisem do *scilaba* (20/II/2003)
- Lab 2 - kontynuacja, funkcje *ode()*-(prosty solver ODE) i *fchamp()*-(rysowanie pól wektorowych) w *scilabie*. Samodzielne zaprogramowanie prostego schematu np midpoint w *Scilabie*. Przykładowy skrypt *scilaba* zawierający otwarty schemat Eulera (6.III.2003)
- Lab 3 - kontynuacja - *fchamp()* i *ode()* w *scilabie*, zapoznanie się z *maplem* - funkcja *dsolve()*, przykładowa sesja *maple'a* . (20.III.2003)
- Lab 4 Badanie rzędu zbieżności schematów w *scilabie* np Eulera otwartego dla równania $y' = -y, y(0) = 1$ dla którego znamy rozwiązanie $y(x) = \exp(-x)$. Należy tymże schematem wyliczać przybliżone rozwiązanie dla ustalonego x np $x=1$ dla połowionych parametrów h np. $h=h_0, h_0/2, h_0/4, h_0/8, \dots, h_0/128$ i obliczać błąd między wyliczonym rozwiązaniem a dokładnym czyli w naszym przykładzie $y(1) = \exp(-1)$, tenże błąd powinien w przybliżeniu maleć 2^p razy przy połowieniu parametru h dla schematu o rzędzie p . (3.IV.2003)
- Lab 5 pakiet *DEsolve* w *maple'u*. Rysowanie portretów fazowych, pól wektorowych przy pomocy funkcji *DEplot*.

²<http://math.rice.edu/~dfield/>

2.1.2 Projekty zaliczeniowe

(Za oddanie po terminie - odpowiednio mniej pktów.)

10pkt Zaimplementować (np. w scilabie) otwarty schemat Rungego rzędu 4 dla równania skalarnego i 2-wymiarowego tzn $y'=f(x,y)$ dla $y \in R^n$ $n=1,2$. Zaprezentować jego działanie pokazując wykresy rozwiązań odpowiednich równań $y'=a*y, y(0)=1$ a - parametr, $y''=y, y(0)=a, y'(0)=b$. (W drugim przypadku wystarczy tylko wykres y - nie trzeba y'). (na Lab 4 - 3.IV.2003).

10pkt (Dwa do wyboru - na ostatni lab)

- Zaimplementować prostą metodę strzałów ze schematem otwartym Rungego rzędu cztery dla równania $-y'' + y = f(x), y(a)=c, y(b)=d$. Wyświetlać wykresy kolejnych strzałów. (najprościej w scilabie - można wykorzystać program z pierwszego projektu).
- Schemat ze zmiennym krokiem całkowania. Opis: [5, rozdział 5.2, str. 151-155], częścią zadania jest samodzielne skorzystanie z literatury.

References

- [1] W. I. Arnold. *Równania różniczkowe zwyczajne*. PWN, Warszawa, 1975.
- [2] N. M. Matwiejew. *Zadania z równań różniczkowych zwyczajnych*. PWN, Warszawa, 1976.
- [3] N. M. Matwiejew. *Metody całkowania równań różniczkowych zwyczajnych*. PWN, Warszawa, 1986.
- [4] Jerzy Ombach. *Wykłady z równań różniczkowych - wspomagane komputerowo - Maple*. Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków, 1999.
- [5] Andrzej Palczewski. *Równania różniczkowe zwyczajne - Teoria i metody numeryczne z wykorzystaniem komputerowego systemu obliczeń symbolicznych*. Wydawnictwo Naukowo Techniczne, Warszawa, 1999.