복합 사다리꼴과 Simpson공식을 이용해 주어진 적분의 근사값 구하기

복합 사다리꼴 & Simpson을 이용한 적분 구하기

과 목 명 : 수치해석
교 수 님 : 고지환 교수님
시 간 : 화4, 목45
학 번 : 20051112
작 성 자 : 이원호
제 출 일 : 2008. 12. 20

목 차

- 프로그램 작성 목적
- 알고리즘
- 프로그램 소스
- 실행 결과
- 결론 및 고찰

프로그램 작성 목적

- 구적법의 정밀도를 높이기 위한 방법으로 복합 수치 적분법을 사용하는데 이 중에 복합 사다리꼴과 Simpson 공식을 이용하여 적분을 구해봄으로써 원리를 이해하고, 적용시키기 위함

- 마디점 개수의 변화에 따라 적분값 변화를 비교 분석하여, 공식의 장・단점과 특징을 알기 위함

알고리즘 – 복합 사다리꼴

- 입력 : a, b, n
- 출력 : DB

$$h = \frac{(b-a)}{n}$$

  - i = 1 부터 n-1까지 다음을 수행한다.

$$x = a + ih$$

$$sum = sum + f(x)$$

  - DB를 출력한다.

$$DB = \frac{h}{2}((f(a)+f(b))+h\times s$$

알고리즘 – 복합 Simpson

- 입력 : a, b, m
- 출력 : DB

$$n = 2 \times m$$

$$h = \frac{(b-a)}{n}$$

  - 아래식을 계산하여 DB를 출력한다.

$$DB = \frac{h}{3}(f(a) + f(b)) + \frac{2h}{3}\sum_{i=1}^{m-1} f(x_{2i}) + \frac{4h}{3}\sum_{i=1}^{m} f(x_{2i-1})$$

프로그램 소스 (1)

```
REAL*8  A, B, DB1, DB2

A=1.D0; B=3.D0  !적분구간

OPEN(UNIT=3, FILE='RESULT.DAT')
WRITE(3,*) '       | N |    | TRAPEZOID |            | SIMPSON |'

DO I = 1, 8
    N = I*20;    M = N/2    !N = 20, 40, 60, 80, 100, 120, 140, 160
    CALL TRAPEZOID(A, B, N, DB1)    !사다리꼴 공식 호출
    CALL SIMPSON(A, B, M, DB2)      !SIMPSON 공식 호출
    WRITE(3,*) N, DB1, DB2
ENDDO

CLOSE(3)
END
```

- MAIN함수부분
  - 복합 사다리꼴과 SIMPSON 공식 SUBROTINE을 호출하여 적분값 출력

프로그램 소스 (2)

```
!-----문제 함수 선언
      FUNCTION F(X)
      REAL*8 F, X
           !1번 F = X / (X**2 + 4)
           !2번 F = X**3 * DEXP(X)
           !3번 F = 2.D0 * DCOS(X)
           !4번 F = X**2 * DEXP(-X**2)
           F = X / (X**2 + 4)
      RETURN
      END

!-----복합 사다리꼴 공식
      SUBROUTINE TRAPEZOID(A, B, N, DB)
      REAL*8 A, B, X, DB, H, F, SUM

      H = (B-A) / N

      SUM = 0.D0
      DO I = 1, N - 1
          X = A + I*H
          SUM = SUM + F(X)
      ENDDO

      DB = H/2.D0 * (F(A)+F(B)) + H*SUM

      RETURN
      END
```

$$h\sum_{i=1}^{n-1} f(x_i)$$

- 적분할 함수 FUNCTION
- 복합 사다리꼴 공식을 이용해 적분을 구함

$$DB = \frac{h}{2}(f(a)+f(b)) + h\sum_{i=1}^{n-1} f(x_i)$$

프로그램 소스 (3)

```
!-----복합 SIMPSON 공식
      SUBROUTINE SIMPSON(A, B, M, DB)
      REAL*8 A, B, X, DB, H, F, SUM1, SUM2

      N = 2*M
      H = (B-A) / N

      SUM1=0.D0
      DO I = 1, M-1
          X = A + 2*I*H
          SUM1 = SUM1 + F(X)
      ENDDO
      SUM1 = 2.D0*H/3.D0 * SUM1

      SUM2=0.D0
      DO I = 1, M
          X = A + (2*I-1)*H
          SUM2 = SUM2 + F(X)
      ENDDO
      SUM2 = 4.D0*H/3.D0 * SUM2

      DB = H/3.D0 * (F(A)+F(B)) + SUM1 + SUM2
      RETURN
      END
```

$$\frac{2h}{3}\sum_{i=1}^{m-1} f(x_{2i})$$

$$\frac{4h}{3}\sum_{i=1}^{m} f(x_{2i-1})$$

- 복합 SIMPSON 공식을 이용해 적분을 구함

$$DB = \frac{h}{3}(f(a)+f(b)) + \frac{2h}{3}\sum_{i=1}^{m-1} f(x_{2i}) + \frac{4h}{3}\sum_{i=1}^{m} f(x_{2i-1})$$

실행 결과 (1)

$$(1)\int_1^3 \frac{x}{x^2+4}dx$$

- 복합사다리꼴과 복합 SIMPSON 공식의 적분값을 비교하면, 복합 SIMPSON 공식이 좀더 정확했다.

실행 결과 (2)

$$(2)\int_{-2}^{2} x^3 e^x dx$$

RESULT.DAT - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

N	TRAPEZOID	SIMPSON
20	20.4102433700580	19.9264269996561
40	20.0434645430609	19.9212049340618
60	19.9753687772011	19.9209226190541
80	19.9515223976575	19.9208750158564
100	19.9404826715088	19.9208619973735
120	19.9344851839059	19.9208573194741
140	19.9308686841145	19.9208553137660
160	19.9285213545478	19.9208543401779

실행 결과 (3)

$$(3)\int_0^2 2x\cos x\,dx$$

RESULT.DAT - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

N	TRAPEZOID	SIMPSON
20	1.81707910529722	1.81859586518588
40	1.81821596393617	1.81859491681582
60	1.81842646212060	1.81859486612623
80	1.81850013418274	1.81859485759826
100	1.81853423341877	1.81859485526797
120	1.81855275635337	1.81859485443097
140	1.81856392506226	1.81859485407217
160	1.81857117396921	1.81859485389803

실행 결과 (4)

$$(4)\int_0^2 x^2 e^{-x^2} dx$$

RESULT.DAT - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

N	TRAPEZOID	SIMPSON
20	0.422541941281371	0.422724884062775
40	0.422679269946519	0.422725046168235
60	0.422704706285739	0.422725054469483
80	0.422713609377277	0.422725055854196
100	0.422717730302074	0.422725056231379
120	0.422719968846423	0.422725056366651
140	0.422721318623945	0.422725056424588
160	0.422722194683839	0.422725056452693

결론 및 고찰

- 복합 사다리꼴 공식과 복합 SIMPSON 공식을 이용해 적분을 구하는 프로그램을 작성하여 실제 손으로 하는 계산과 달리 컴퓨터로 계산시 방법의 차이를 확인 할 수 있었다.
- 프로그래밍시 SUBROTINE을 사용하여 다른 곳에서 적분 계산이 필요할 때 편하게 쓸 수 있어 범용성을 높일 수 있었다.