



Bilgisayar Müh. için Diferansiyel Denklemler

BLM2642

Prof. Dr. Sırma YAVUZ

Ödev 2: 4. Derece Runge-Kutta Metodu ile Diferansiyel Denklemleri Çözme

Mehmet Ali Duran

21011090

ali.duran@std.yildiz.edu.tr

Ödevin Konusu:

N terimli sabit katsayılı doğrusal adi diferansiyel denkleminin Runge Kutta-4 yöntemiyle sayısal çözümünü C dilinde kodlayınız.

Girdiler:

Denklemin terim sayısını ifade eden n değeri ve bu n değerine göre diferansiyel denklemin katsayıları kullanıcıdan alınmalıdır. Ayrıca, sayısal çözüm için denklem çözümü sonrasında elde edilecek sayısal değer için bağımsız değişken değeri (t veya x) kullanıcıdan alınmalıdır.

Denklemleri modellerken mutlaka struct kullanınız.

Çıktı:

Runge Kutta -4 ile elde edilen çözümünün t veya x bağımsız değişkenine göre yaklaşık çözümü ekranda gösterilmelidir. Ayrıca iterasyonlarda elde edilen yaklaşım sonuçları aşamalı olarak ekrana yazdırılmalıdır. Raporunuzda olacak test örneklerinde çözümü bilinen denklemleri kullanınız, bu sayede üretilen nümerik çözümle gerçek çözüm arasındaki mutlak hata değerini ekranda gösteriniz.

Açıklamalar

Bu ödevde Runge-Kutta4 metodu ile diferansiyel denklemlerin çözümünü iteratif bir şekilde yapan C programının açıklaması vardır. Programda denklemler bir struct yapısı ile tutulmuştur ve bu yapı denkleme ait $N \rightarrow$ terim sayısını, $\text{coefficient} \rightarrow$ terimlerin katsayılarını, $\text{order} \rightarrow$ terimlerin derecelerini, $\text{var} \rightarrow$ o anki değişkenin x mi y mi olduğunu tutmaktadır. Ayrıca malloc() fonksiyonu için stdlib.h ve pow() fonksiyonu için math.h kütüphaneleri kullanılmıştır.

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

struct equation{
    int N;
    float *coefficient;
    int *order;
    int *var;
};
```

Burada şöyle bir tasarım yapılmıştır. y' ifadesini alınca karşıda $f(x, y)$ yani x ve y 'lerden oluşan terimler kalıyor, sabit katsayılı dediği için $x.y$ gibi ifadeler de bulunmaması gerekir. Bu durumda örnek olarak $x^2 + y$ gibi bir ifade olabilir bu ifadede x ve y 'yi yerine koyup işlem yapmak istediğimizde o an işlem yaptığımız ifadenin x mi y mi olduğunu bilmemiz gerekiyor. Bunun için de ayrıca bir var dizisi tutularak bu dizi içinde ifade x ise 1, y ise 0 olarak tutuluyor. İlerde değer hesaplama kısmında bu bilgiden yararlanılarak hangi değişken değerinin kullanılacağı seçiliyor. Bunu da fonksiyonun içinde basit bir if kontrolü ile yapıyoruz.

Kodda kullanılan fonksiyonlar :

```
float calculate_new_iter(float h, float xi, float current_value, struct equation func, int i);
float calculate_derivative(float xi, float yi, struct equation func);
float calculate_k(float xi, float yi, struct equation func);
```

Kısaca işlevleri ve içerikleri şöyle:

`calculate_new_iter`: Bu fonksiyon parametre olarak aldığı `xi` ve `yi(current_value)` değerleri ve `h` ile o anki değer üzerinden yeni değeri hesaplar.

`current_value` başlangıçtaki değerdir (yani `y0`), `y(x0) = y0`. Bu formül Runge-Kutta 4'ün kendisinden gelmektedir.

```
float calculate_new_iter(float k1, float k2, float k3, float k4, float h, float current_value){
    float result;

    result = current_value + (h/6)*(k1 + (2*k2) + (2*k3) + k4);

    return result;
}
```

`calculate_derivative`: Bu fonksiyon `y'` fonksiyonunda `x` ve `y'` yi yerine koyarak sonucu döndürür.

```
float calculate_derivative(float xi, float yi, struct equation func){
    float result=0;
    int i;

    for(i=0;i<func.N;i++){
        if(func.var[i] == 1){
            result += func.coefficient[i]*pow(xi, func.order[i]);
        }else{
            result += func.coefficient[i]*pow(yi, func.order[i]);
        }
    }

    //test function
    //result += xi + (yi/xi);

    return result;
}
```

calculate_k: Bu fonksiyon calculate_derivative fonksiyonu üzerinden k değerlerini hesaplar. Burada k değerleri için gelen parametreler farklıdır.

```
float calculate_k(float xi, float yi, struct equation func){  
    float result;  
    result = calculate_derivative(xi, yi, func);  
    return result;  
}
```

```
k1 = calculate_k(xi, current_value, func);  
k2 = calculate_k(xi+(h/2.0), current_value+((k1*h)/2.0), func);  
k3 = calculate_k(xi+(h/2.0), current_value+((k2*h)/2.0), func);  
k4 = calculate_k(xi+h, current_value+(k3*h), func);
```

Kodumuz gerekli girdileri aldıktan sonra her iterasyonun sonucunu ve o iterasyon için hata değerini ekrana basar.

Şimdi de 4 örnek üzerinden kodun çıktıları gösterilecektir.

Örnek-1

Find $y(0.2)$ for $y'=(x-y)/2$, $x_0=0, y_0=1$, with step length 0.1 using Runge-Kutta 4 method (1st order derivative)

$$y' = 0.5x - 0.5y \quad x_0=0, \quad y_0=1, \quad h=0.1,$$

Real $Y(0.1) = 0.9536882735$

Real $Y(0.2) = 0.9145122541$

```
please enter the term number N: 2
Please enter 1.th terms if x or y (1 for x, 0 for y): 1
Please enter the 1.th terms coefficient: 0.5
Please enter the 1.th terms order: 1
Please enter 2.th terms if x or y (1 for x, 0 for y): 0
Please enter the 2.th terms coefficient: -0.5
Please enter the 2.th terms order: 1
please enter the initial xi: 0
please enter the initial yi: 1
please enter the target x: 0.2
please enter the h value: 0.1
```

```
counted iter number is 2
please enter the 1.th iteration real value for y(0.100): 0.9536882735
please enter the 2.th iteration real value for y(0.200): 0.9145122541

k1:-0.500000    k2:-0.462500    k3:-0.463437    k4:-0.426828
1.th iteration result for y(0.100) is 0.953688          error:0.000000

k1:-0.426844    k2:-0.391173    k3:-0.392065    k4:-0.357241
2.th iteration result for y(0.200) is 0.914512          error:0.000000
```

Örnek-2

$Y' = -2x - y$, $x_0 = 0$, $y_0 = -1$, $h = 0.1$, $y(0.5) = ?$

Real $Y(0.1) = -0.9145122541$

Real $Y(0.2) = -0.8561922592$

Real $Y(0.3) = -0.822454662$

Real $Y(0.4) = -0.8109601381$

Real $Y(0.5) = -0.8195919791$

```
please enter the term number N: 2
Please enter 1.th terms if x or y (1 for x, 0 for y): 1
Please enter the 1.th terms coefficient: -2
Please enter the 1.th terms order: 1
Please enter 2.th terms if x or y (1 for x, 0 for y): 0
Please enter the 2.th terms coefficient: -1
Please enter the 2.th terms order: 1
please enter the initial xi: 0
please enter the initial yi: -1
please enter the target x: 0.5
please enter the h value: 0.1
```

```
counted iter number is 5
please enter the 1.th iteration real value for y(0.100): -0.9145122541
please enter the 2.th iteration real value for y(0.200): -0.8561922592
please enter the 3.th iteration real value for y(0.300): -0.822454662
please enter the 4.th iteration real value for y(0.400): -0.8109601381
please enter the 5.th iteration real value for y(0.500): -0.8195919791
```

```
k1:1.000000    k2:0.850000    k3:0.857500    k4:0.714250
1.th iteration result for y(0.100) is -0.914513    error:0.000000
```

```
k1:0.714513    k2:0.578787    k3:0.585573    k4:0.455955
2.th iteration result for y(0.200) is -0.856193    error:0.000000
```

```
k1:0.456193    k2:0.333383    k3:0.339524    k4:0.222240
3.th iteration result for y(0.300) is -0.822455    error:0.000001
```

```
k1:0.222455    k2:0.111332    k3:0.116889    k4:0.010766
4.th iteration result for y(0.400) is -0.810961    error:0.000001
```

```
k1:0.010961    k2:-0.089587    k3:-0.084560    k4:-0.180583
5.th iteration result for y(0.500) is -0.819593    error:0.000001
```

Örnek-3

$Y' = -y$, $x_0 = 0$, $y_0 = 1$, $h = 0.1$, $y(0.2) = ?$

Real $Y(0.1) = 0.904837418$

Real $Y(0.2) = 0.8187307531$

```
please enter the term number N: 1
Please enter 1.th terms if x or y (1 for x, 0 for y): 0
Please enter the 1.th terms coefficient: -1
Please enter the 1.th terms order: 1
please enter the initial xi: 0
please enter the initial yi: 1
please enter the target x: 0.2
please enter the h value: 0.1
```

```
counted iter number is 2
please enter the 1.th iteration real value for y(0.100): 0.904837418
please enter the 2.th iteration real value for y(0.200): 0.8187307531

k1:-1.000000    k2:-0.950000    k3:-0.952500    k4:-0.904750
1.th iteration result for y(0.100) is 0.904837          error:0.000000

k1:-0.904837    k2:-0.859596    k3:-0.861858    k4:-0.818652
2.th iteration result for y(0.200) is 0.818731          error:0.000000
```


Örnek-4

$$y' = x^2 + y \quad x_0 = 0, \quad y_0 = 1, \quad h = 0.1, \quad y(0.2) = ?$$

Real $Y(0.1) = 1.105512754$

Real $Y(0.2) = 1.224208274$

```
please enter the term number N: 2
Please enter 1.th terms if x or y (1 for x, 0 for y): 1
Please enter the 1.th terms coefficient: 1
Please enter the 1.th terms order: 2
Please enter 2.th terms if x or y (1 for x, 0 for y): 0
Please enter the 2.th terms coefficient: 1
Please enter the 2.th terms order: 1
please enter the initial xi: 0
please enter the initial yi: 1
please enter the target x: 0.2
please enter the h value: 0.1
```

```
counted iter number is 2
please enter the 1.th iteration real value for y(0.100): 1.105512754
please enter the 2.th iteration real value for y(0.200): 1.224208274

k1:1.000000    k2:1.052500    k3:1.055125    k4:1.115512
1.th iteration result for y(0.100) is 1.105513    error:0.000000

k1:1.115513    k2:1.183788    k3:1.187202    k4:1.264233
2.th iteration result for y(0.200) is 1.224208    error:0.000000
```