

**T.C.  
PAMUKKALE ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**NÜMERİK İNTEGRASYON**

—

**Soldan Dikdörtgen – Sağdan Dikdörtgen – Mid Point – Yamuklar Yöntemi  
– Simpson 1/3 – Simpson 3/8 – Weddle – Boole**

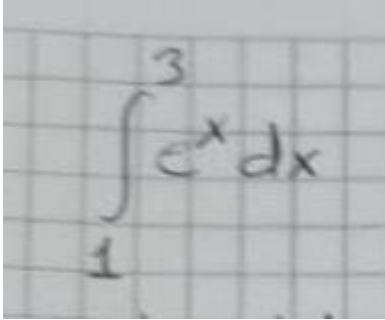
**SAYISAL ÇÖZÜMLEME DERSİ  
2021 - 2022 GÜZ YARIYILI**

**14.12.2021**

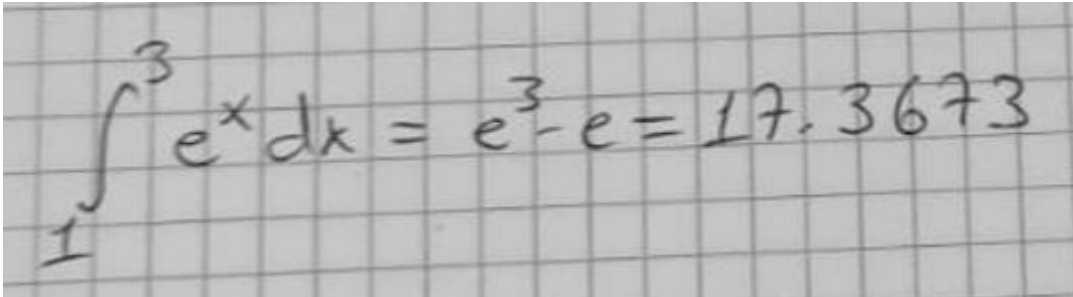
**Hazırlayan**

**Cenk ŞENGÜNLÜ  
20253605**

**Numerik İntegrasyon metotları için kullandığımız integral:**


$$\int_1^3 e^x dx$$

**Bu integral için n = 30;**


$$\int_1^3 e^x dx = e^3 - e = 17.3673$$

**Ve işlemler sonunda çıkan sonuç 17.3673 değerine olabildiğince yakın olmalıdır.**

**Uygulayacağımız Metotlar;**

- a)** Soldan Dikdörtgen
- b)** Sağdan Dikdörtgen
- c)** Mid Point
- d)** Yamuklar Yöntemi
- e)** Simpson 1/3
- f)** Simpson 3/8
- g)** Weddle
- h)** Boole

Numerik İntegrasyon metotlarını uygulamak için JavaScript dilinde yazdığım kod;

```
1  /*
2      UYGULAYACAĞIMIZ METOTLAR
3
4      a) Soldan Dikdörtgen
5      b) Sağdan Dikdörtgen
6      c) Mid Point
7      d) Yamuklar Yöntemi
8      e) Simpson 1/3
9      f) Simpson 3/8
10     g) Weddle
11     h) Boole
12 */
13
14 // Linspace Fonksiyonu
15 function linspace(start, stop, cardinality){
16     let arr = [];
17     let step = (stop - start) / (cardinality - 1);
18     for(let i = 0; i < cardinality; i++){
19         arr.push(start + (step * i));
20     }
21     return arr;
22 }
23
24 // Global Değişkenler
25 let a = 1; // Alt Değer
26 let b = 3; // Üst Değer
27 let n = 30; // Aralık
28 let h = (b - a) / n;
29 let arr = linspace(a, b, n); /* Linspace Fonksiyonu ile a ve b arasındaki
30                               n sayıda eşit sayıdan oluşan array */
31
32 // İntegral Fonksiyonu
33 function f(x){
34     return Math.exp(x);
35 }
36
```

```
36
37 // a) Soldan Dikdörtgen
38 function soldanDikdortgen(){
39     let sum = 0;
40
41     for(let i = 1; i < n; i++) {
42         sum += f(arr[i - 1]) * h;
43     }
44
45     console.log(`Soldan Dikdörtgen: ${sum.toFixed(4)}`);
46 }
47 soldanDikdortgen();
48
49 // b) Sağdan Dikdörtgen
50 function sagdanDikdortgen(){
51     let sum = 0;
52
53     for(let i = 1; i < n; i++) {
54         sum += f(arr[i]) * h;
55     }
56
57     console.log(`Sağdan Dikdörtgen: ${sum.toFixed(4)}`);
58 }
59 sagdanDikdortgen();
60
61 // c) Mid Point
62 function midPoint(){
63     let sum = 0;
64
65     for(let i = 0; i < n - 1; i++) {
66         sum += (f((arr[i] + arr[i + 1]) / 2) * (arr[i + 1] - arr[i]));
67     }
68
69     console.log(`Mid Point: ${sum.toFixed(4)}`);
70 }
71 midPoint();
72
```

```

72
73 // d) Yamuklar Yöntemi
74 function yamuklarTeoremi(){
75     let sum = 0;
76     let head;
77     let end;
78
79     for(let i = 0; i < n - 1; i++) {
80         head = arr[i];
81         end = arr[i + 1];
82         sum += (end - head) * ((f(head) + f(end)) / 2);
83     }
84
85     console.log(`Yamuklar Yöntemi: ${sum.toFixed(4)}`);
86 }
87 yamuklarTeoremi();
88
89 // e) Simpson 1/3
90 function simpsonOneThree(){
91     let res = f(a) + f(b);
92     let lim = n / 2;
93     let oddSum = 0;
94     let evenSum = 0;
95
96     for(let i = 1; i <= lim; i++){
97         oddSum += f(a + (2 * i - 1) * h);
98     }
99     oddSum *= 4;
100
101     for(let j = 1; j < lim; j++){
102         evenSum += f(a + (2 * j) * h);
103     }
104     evenSum *= 2;
105
106     res += oddSum + evenSum;
107     res *= (h / 3);
108
109     console.log(`Simpson 1/3: ${res.toFixed(4)}`);
110 }
111 simpsonOneThree();
112

```

```
112
113 // f) Simpson 3/8
114 function simpsonThreeEight() {
115     let sum = f(a) + f(b);
116
117     for(let i = 1 ; i < n ; i++)
118     {
119         if(i % 3 === 0){
120             sum += 2 * f(a + i * h);
121         } else{
122             sum += 3 * f(a + i * h);
123         }
124     }
125
126     console.log(`Simpson 3/8: ${((3 * h / 8) * sum).toFixed(4)}`);
127 }
128 simpsonThreeEight();
129
130
131 // g) Simpson 3/8
132 function weddle(){
133     let h = (b - a) / 6;
134     let sum = 0;
135
136     sum += (((3 * h) / 10) *
137         (f(a) + f(a + 2 * h) + 5 * f(a + h) +
138         6 * f(a + 3 * h) + f(a + 4 * h) +
139         5 * f(a + 5 * h) + f(a + 6 * h)));
140
141     console.log(`Weddle: ${sum.toFixed(4)}`);
142 }
143 weddle();
144
```

```

144
145 // h) Boole
146 function booleYontemi(){
147     let sum = 0;
148     n = 4;
149     h = (b - a) / n;
150
151     let bl = (7 * f(a) + 32 * f(a + h)
152             + 12 * f(a + 2 * h)
153             + 32 * f(a + 3 * h)
154             + 7 * f(a + 4 * h))
155             * 2 * h / 45;
156
157     sum += bl;
158
159     console.log(`Boole: ${sum.toFixed(42)}`);
160 }
161 booleYontemi();

```

Yazdığım fonksiyonların çıktıları:

Soldan Dikdörtgen: 16.2161

Sağdan Dikdörtgen: 17.3739

Mid Point: 17.3638

Yamuklar Yöntemi: 17.3741

Simpson 1/3: 17.3673

Simpson 3/8: 17.3673

Weddle: 17.3673

Boole: 17.3678

Çıktıların bir kısmı 17.3673 değerine eşitken kalanları bu değere oldukça yakın.

