درس سوم: روش هم‌مکانی

۲۶ اردیبهشت ۱۴۰۲

برای حل معادله انتگرال فردهلم

$$u(x) = g(x) + \int_a^b k(x,y)u(y)\mathrm{d}y, \qquad (۱)$$

به روش هم‌مکانی صورت زیر عمل می‌کنیم. ابتدا یک پایه از توابع با بعد متناهی انتخاب می‌کنیم. یک انتخاب ساده عبارت است از

$$X_n = \mathrm{span}\{1, x, \ldots, x^{n-1}\}.$$

می‌توان به طور کلی X_n را به صورت زیر در نظر گرفت:

$$X_n = \mathrm{span}\{\phi_1(x), \phi_2(x), \ldots, \phi_n(x)\}.$$

هر عضو X_n به صورت ترکیب خطی از $\phi_i(x)$ است. حال فرض کنید تقریب جواب معادلهٔ (۱) عضوی از X_n باشد، در این صورت

$$u_n(x) = \sum_{i=1}^{n} c_i\phi_i(x).$$

پس با جایگذاری جواب تقریب در معادلهٔ انتگرال خواهیم داشت:

$$\sum_{i=1}^{n} c_i\{\phi_i(t) - \int_0^1 k(t,s)\phi_i(s)\mathrm{d}s\} \approx g(t), \qquad (۲)$$

عبارت بالا را هم‌مکانی می‌کنیم، یعنی نقاط دلخواهی مانند $\{t_i\}_{i=1}^{n}$ را در بازهٔ $[0,1]$ در نظر می‌گیریم. این نقاط را نقاط هم‌مکانی می‌نامیم. ضرایب نامعلوم c_i را چنان تعیین می‌کنیم که عبارت (۲) در نقاط به طور دقیق برقرار باشد. این ایده منجر به دستگاه معادلات زیر می‌شود:

$$\sum_{i=1}^{n} c_i\{\phi_i(t_j) - \int_0^1 k(t_j,s)\phi_i(s)\mathrm{d}s\} = g(t_j), \quad j \in 1:n \qquad (۳)$$

با حل این دستگاه، ضرایب c_i بدست می‌آید. در ادامه کدهای این روش را مشاهده می‌کنید:

فراموش نکنید که کتابخانه‌های لازم را فراخوانی کنید!

```
using QuadGK, LinearAlgebra, Plots, FastGaussQuadrature


right_hand(x) = 2*exp(x)+(1-exp(1+x))/(1+x)
kernel(x,y)= exp(x*y)
exact(x) = exp(x)

#basic functions

function psi(x,i)
return x^(i-1)
end

#collocation points
points = LinRange(0, 1, 100)
```

کد زیر را برای انتگرال‌گیری عددی استفاده می‌کنیم. البته اینجا شما می‌توانید از روش‌های دیگر انتگرال‌گیری عددی نیز استفاده کنید:

```
function gausss(f::Function, a, b, n)
    x,w =  gausslegendre(n)
    w = 0.5*(b-a)*w # new weights after changing of variable
    s = 0
    for i in 1:n
    s += w[i]*f(0.5*(b-a)*x[i]+ 0.5*(b+a))
end

return s
end
```

```
function integrationpart(t, n)
    points = LinRange(0, 1, n)
    n = size(points)[1]
    aa = zeros(n)
    bb = zeros(n)
    for i in 1:n
        f1(s) = kernel(t,s)*psi(s,i)
        qa = gausss(s->f1(s), 0, 1, 7)
        qb = quadgk(f1, 0, 1)[1]
        aa[i] = qa
        bb[i] = qb
    end
    return aa
end
```

```
function coefficientmatrix(points)
    n = length(points)
    A = zeros(n,n)
```

```
    for i in 1:n for j in 1:n
        A[i,j] = 2*psi(points[j],i)-integrationpart(points[j], n)[i]
    end end
    return A'
end
```

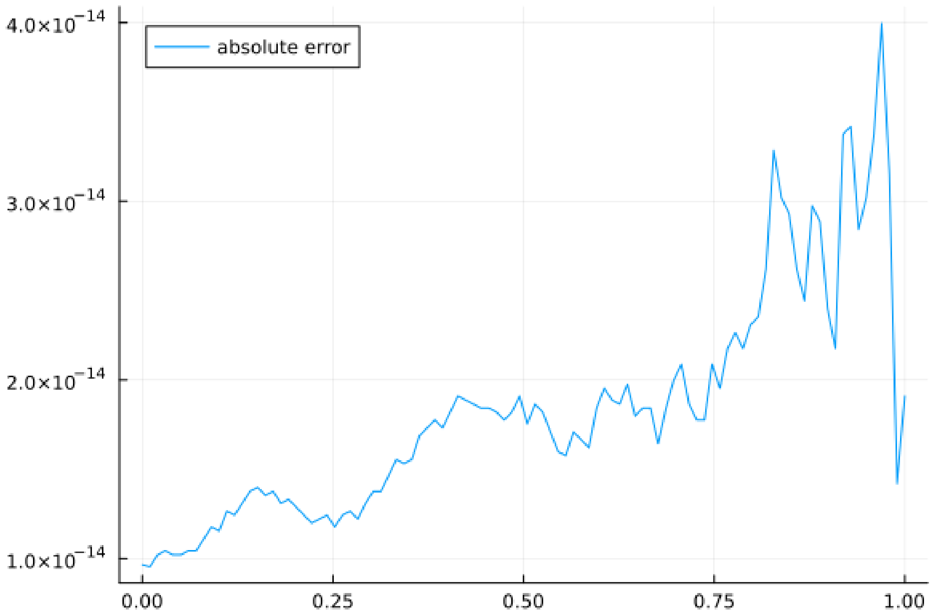
```
function RH(points)
    n = length(points)
    b = zeros(n)
    for i in 1:n
        b[i] = right_hand(points[i])
    end
    return b
end
```

```
function collocation(points)
    A = coefficientmatrix(points)
    b = RH(points)
return A\b
end
```

```
z = collocation(points)
function app(x)
    sum = 0
    for i in 1:length(z)
        sum += z[i]*psi(x,i)
    end
    return sum
end

using Plots
x = LinRange(0,1,100)
y1 = app.(x)
#plot(x, y1, label = "approx")
#plot!(x, exact.(x), label = "exact")
plot(x, abs.(y1-exact.(x)), label = "absolute error")
```

خروجی نهایی را در نمودار زیر مشاهده می‌کنیم:


absolute error
4.0×10^{−14}
3.0×10^{−14}
2.0×10^{−14}
1.0×10^{−14}
0.00
0.25
0.50
0.75
1.00