```
Activating project at `~/.julia/environments/v1.8`
```

Метод замены ядра на вырожденное

$$\varphi(x) - 0.2 * \int\limits_0^1 \frac{1}{10 - x * y} \varphi(y) \mathbb{d}y = 1 + x^2$$

```
(0, 1)
```

```
A, B = 0, 1
```

```
λ = 0.2
```

```
λ = 0.2
```

```
K (generic function with 1 method)
```

```
K(x, y) = 1 / (10 - x*y)
```

```
f (generic function with 1 method)
```

```
f(x) = 1 + x^2
```

Разложение в ряд Тейлора в (1/2, 1/2)

```
r = 4
```

```
r = 4
```

```
(x, y)
```

```
@syms x, y
```

$\mathbb{K}$ =

$$\frac{1}{-\mathbb{x}\mathbb{y} + 10}$$

```
𝕂 = 1 / (10 - 𝕩 * 𝕪)
```

$\mathbb{f}$ =

$$\mathbb{x}^2 + 1$$

```
𝕗 = 1 + 𝕩^2
```

$$\frac{\mathbb{y}}{(-\mathbb{x}\mathbb{y} + 10)^2}$$

```
diff(𝕂, 𝕩)
```

! (generic function with 1 method)

```
!(n::Int) = n == 0 ? 1 : n * !(n-1)
```

$\mathbb{X}$ =

$$7.09344064112999 \cdot 10^{-7}\mathbb{x}^4 + 8.58306317576729 \cdot 10^{-5}\mathbb{x}^3\mathbb{y} - 4.43340040070624 \cdot 10^{-5}\mathbb{x}^3 + 0$$

```
𝕏 = sum(
    diff(diff(𝕂, 𝕩, p), 𝕪, r-p).subs(Dict(
        𝕩 => .5, 𝕪 => .5
    )) * (𝕩-.5)^p * (𝕪-.5)^(r-p) / !(p) / !(r-p)
    for p in 0:r
) |> expand |> simplify
```

$$8.58306317576729 \cdot 10^{-5}\mathbb{y}^3 - 0.0013779008445395\mathbb{y}^2 + 0.0013779008445395\mathbb{y} - 0.00035555$$

```
𝕏.coeff(𝕩)
```

prods =

[($\mathbb{y}^4$, 0, 4), ($\mathbb{x}\mathbb{y}^3$, 1, 3), ($\mathbb{y}^3$, 0, 3), ($\mathbb{x}^2\mathbb{y}^2$, 2, 2), ($\mathbb{x}\mathbb{y}^2$, 1, 2), ($\mathbb{y}^2$, 0, 2), ($\mathbb{x}^3\mathbb{y}$, 3, 1), ($\mathbb{x}$

```
prods = [
    (𝕩^p * 𝕪^q, p, q)
    for p in 0:r, q in 0:r if p+q ≤ 4
] |> reverse
```

```
[7.09344e-7, 8.58306e-5, -4.4334e-5, 0.00124915, -0.0013779, 0.000377726, 8.58306e-5, -0.0
```

```julia
begin
    buf = 𝕏
    coeffs = []
    for p in prods[1:end-1] .|> first
        push!(coeffs, buf.coeff(p) |> N)
        global buf -= buf.coeff(p) * p
    end
    push!(coeffs, buf |> N |> BigFloat)
    coeffs
end
```

```
terms = OrderedCollections.OrderedDict{Tuple{SymPy.Sym, Int64, Int64}, AbstractFloat}(
    (y^4, 0, 4) ⇒ 7.09344e-7
    (xy^3, 1, 3) ⇒ 8.58306e-5
    (y^3, 0, 3) ⇒ -4.4334e-5
    (x^2y^2, 2, 2) ⇒ 0.00124915
    (xy^2, 1, 2) ⇒ -0.0013779
    (y^2, 0, 2) ⇒ 0.000377726
    (x^3y, 3, 1) ⇒ 8.58306e-5
    (x^2y, 2, 1) ⇒ -0.0013779
    (xy, 1, 1) ⇒ 0.0013779
    (y, 0, 1) ⇒ -0.000355559
    (x^4, 4, 0) ⇒ 7.09344e-7
    (x^3, 3, 0) ⇒ -4.4334e-5
    (x^2, 2, 0) ⇒ 0.000377726
    (x, 1, 0) ⇒ -0.000355559
    (1, 0, 0) ⇒ 8.88897e-05
)
```

```julia
terms = OrderedDict(prods .=> coeffs)
```

Каждый член суммы представим в виде $c_{p+q} * x^{p-1} * y^{q-1}$, тогда $c_{p+q} = a_{p+q} * p * q = (a_{p+q} * p^2/q) * (a_{p+q} * q^2/p)$. Функции $\alpha(x), \beta(y)$ примут вид:

$$\begin{cases} \alpha(x) = a_{p+q} * q * x^{p-1} \\ \beta(y) = p * y^{q-1} \end{cases}$$

$([1.58614154679264 \cdot 10^{-6},\ 0.000121382843498752\mathbb{x},\ -8.86680080141249 \cdot 10^{-5},\ 0.0012491548969$

```
begin
    α, β = [], []
    term_keys = collect(keys(terms))
    for key in term_keys[1:end-1]
        push!(
            α,
            terms[key] * (key[3]+1) * x^key[2] / sqrt((key[2]+1) * (key[3]+1))
        )
        push!(
            β,
            (key[2]+1) * y^key[3] / sqrt((key[2]+1) * (key[3]+1))
        )
    end
    α[end] += terms[term_keys[end]] / β[end]
    α, β
end
```

$840434497101 \cdot 10^{-19}\mathbb{x}^2\mathbb{y} + 1.0842021724855 \cdot 10^{-19}\mathbb{x}^2 - 1.35525271560688 \cdot 10^{-20}\mathbb{x}\mathbb{y}^3 - 2.168$

```
# проверка на маленьковость ошибки
sum(
    α[i] * β[i]
    for i in 1:14
) - X
```

Построение системы и вычисление A_i

```
B = 14×14 Matrix{Float64}:
  1.0          -5.60786e-8  -3.96535e-8  -1.05743e-7  …  -5.49456e-7   -4.48629e-7
 -1.80947e-6    0.999997    -2.42766e-6  -6.06914e-6     -2.10241e-5   -1.71661e-5
  1.58614e-6    3.13489e-6   1.0          5.9112e-6       3.07155e-5    2.50791e-5
 -1.59611e-5   -2.94429e-5  -2.08192e-5   0.99995        -0.00014424   -0.000117771
  2.51569e-5    4.77319e-5   3.37515e-5   8.43788e-5      0.000292297   0.000238659
 -1.17034e-5   -2.31309e-5  -1.6356e-5   -4.3616e-5   …  -0.000226635  -0.000185047
 -6.78551e-7   -1.22615e-6  -8.6702e-7   -2.02305e-6     -5.25603e-6   -4.29153e-6
  1.43754e-5    2.65177e-5   1.87509e-5   4.50021e-5      0.00012991    0.000106071
 -2.05405e-5   -3.89729e-5  -2.7558e-5   -6.8895e-5      -0.000238659  -0.000194865
  8.995e-6      1.77779e-5   1.25709e-5   3.35224e-5      0.000174187   0.000142223
 -3.15264e-9   -5.60786e-9  -3.96535e-9  -9.06367e-9  …  -2.19782e-8   -1.79451e-8
  2.47835e-7    4.47841e-7   3.16671e-7   7.389e-7        1.91972e-6    1.56744e-6
 -2.78652e-6   -5.1402e-6   -3.63467e-6  -8.7232e-6      0.999975     -2.05608e-5
  2.62354e-6    4.88893e-6   3.457e-6     8.3806e-6       2.17734e-5    1.00002
```

```julia
B = -λ * [
    integrate(
        α[i] * β[j].subs(Dict(y=>x)),
        (x, A, B)
    ) |> N |> Float64
    for i in 1:length(α), j in 1:length(β)
] + I
```

```
F =  Float64[
       1:  0.15333
       2:  0.294628
       3:  0.208333
       4:  0.533333
       5:  0.435465
       6:  0.30792
       7:  1.06066
       8:  0.918559
       9:  0.75
       10:  0.53033
       11:  2.98142
       12:  2.66667
       13:  2.3094
       14:  1.88562
       ]
```

```julia
F = [
    integrate(
        f * β[j].subs(Dict(y=>x)),
        (x, A, B)
    ) |> N
    for j in 1:length(β)
]
```

```
A =   Float64[
        1:  0.153337
        2:  0.294893
        3:  0.207961
        4:  0.535191
        5:  0.431778
        6:  0.310667
        7:  1.06073
        8:  0.916886
        9:  0.75301
        10:  0.528219
        11:  2.98142
        12:  2.66664
        13:  2.30973
        14:  1.88533
        ]
```

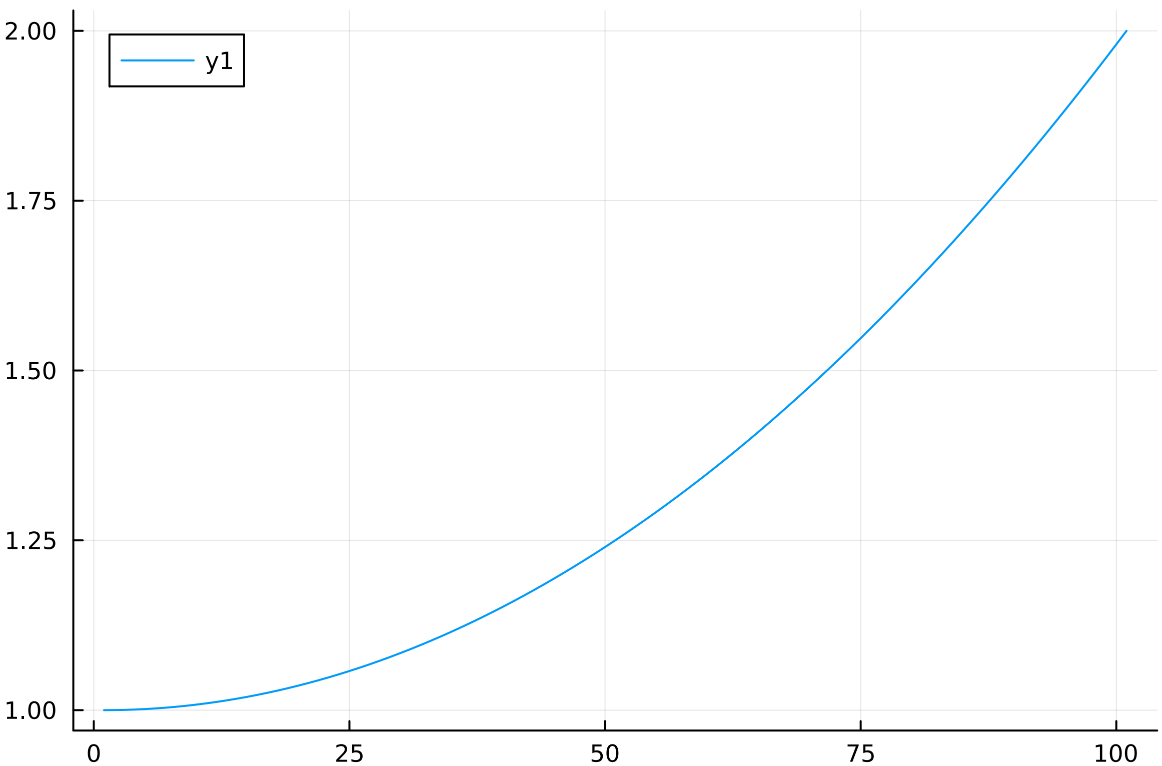
```julia
A = B \ F
```

$\mathbb{x} + 1.0000075896424289085933768414427856714542050737143828923465322173623961673 66$

```julia
𝕗 + λ * sum(A[i] * α[i] for i in 1:14)
```

```
φ (generic function with 1 method)
```

```julia
φ(x::Number) = (𝕗 + λ * sum(A[i] * α[i] for i in 1:14)).subs(Dict(𝕩 => x)) |> N
```




```julia
φ.(0:.01:1) |> plot
```