

Lecția 11. Probleme de calcul cu aproximare data eps

<http://www.cs.ubbcluj.ro/admitere/nivel-licenta/tematica-admitere-facultate/>

1. Să se calculeze valoarea lui $\sqrt{x}$ fără a utiliza funcții specifice.

Indicație de rezolvare:

Vom nota $\sqrt{x} = y$, unde y este rezultatul dorit, și vom citi inițial precizia eps (spre exemplu eps=0.001). Ecuația $\sqrt{x} = y$ prin ridicare la putere devine: $x=y^2 \Rightarrow y^2-x=0$. Trebuie să determinăm un interval $[a,b]$, astfel încât $b-a < \text{eps}$, care să conțină soluția ecuației. Verificarea existenței soluției în $[a,b]$ conduce la îndeplinirea relației $f(a)*f(b) < 0$, deoarece semnul funcției se schimbă în jurul soluției.

Algoritmul este:

```
citeste x, eps
┌daca x<1 atunci //inițializarea intervalului în care vom căuta soluția
│   a=x           // solutia apartine intervalul [x,1]
│   b=1
│altfel
│   a=1           // solutia apartine intervalul [1,x]
│   b=x
└─┘
┌cat timp b-a>eps executa
│   m=(b+a)/2      //injumatatim intervalul de cautare
│   ┌daca (a*a-x)*(m*m-x)<0 atunci //solutia va fi [a, m]
│   │   b=m
│   │altfel
│   │   a=m          //solutia va fi [m, b]
│   └─┘
└─┘
scrie (a+b)/2
```

2. Fie $f:[a,b] \rightarrow \mathbb{R}$ o funcție continuă cu proprietatea $f(a)*f(b) < 0$. Scrieți un subalgoritm care determină o rădăcină a funcției f , din $[a,b]$, cu aproximarea eps.

Exemplu: $f(x) = x*x*x - x - 2$ sau $f(x) = x*x + 10 * x + 3$;

Exemple

Date de intrare				Rezultat
a	b	eps	f	
-8	8	0.001	X^3-x-2	1.521
-8	8	0.00001	X^3-x-2	1.52138
-2	2	0.001	$X^2+10x+3$	-0.310059
-2	2	0.00001	$X^2+10x+3$	-0.309582
-3	5	0.001	$X^2+10x-3$	0.291504
-3	5	0.1	$X^2+10x-3$	0.28125

```
// Calculeaza cu aprox eps, radacina ecuatiei x^3-x-2 care apartine in intervalul [a,b].
#include <iostream>
using namespace std;
```

```

float a,b,eps;
//desc: Functia continua.
//in: valoarea pentru care calculez functia
//out: rezultatul functiei in punctul dat
float F(float x)
{
    return x*x+10*x+3;
}
//desc: Calculeaza cu aprox eps, radacina ecuatiei x^3-x-2 care apartine in intervalul [a,b].
//in: a,b,eps. Intervalul care contine solutia si aproximarea ceruta.
//out: radacina ecuatiei.
float Radacina(float a,float b,float eps)
{
    while(b-a>eps)
    {
        float m=(a+b)/2;
        if (F(a)*F(m)<0)
            b=m;
        else
            a=m;
    }
    return (a+b)/2;
}
int main()
{
    cin>>a>>b>>eps;
    cout<<Radacina(a,b,eps);
    return 0;
}

```

3. Cu metoda dedusă, să se calculeze $\sqrt[n]{M}$. Indicație. Se va lua $f(x) = x^n - M$, iar intervalul $[a,b]$, se înlocuiește cu $[0,M]$.
4. Se cunoaște seria convergentă:

$$\sum_{n=1}^{\infty} \left(\frac{1}{3*n-2} * \frac{1}{3*n-1} * \frac{1}{3*n} \right)$$

pana cand $s_{n+1}-s_n < \text{eps}$.

```

#include <iostream>
using namespace std;
//descriere : calculez valoare expresiei din problema 4 fisierul aproxmari
//           cu precizia eps
//int: eps
//out : valoarea expresiei
float expresie(float eps)
{float n=1;
  float t=1/6.0; // 6!= 6.0 trebuie facut unul din termeni real
  float e=0;
  while (t>=eps)
  {n++;
    t=(1/(3*n-2))*(1/(3*n-1))*(1/(3*n));
  }
}

```

```

        e=e+t;
    }
    return e;
}
int main()
{float eps;
cout<<"eps=";
cin>>eps;
cout <<expresie(eps)<< endl;
    return 0;
}

```

Să se determine valoarea lui s_n , astfel încât suma să fie îndeplinită relația $|s_n - s_{n-1}| < eps$.

5. Se cunoaște seria convergentă:

$$\sum_{n=1}^{\infty} (-1)^{n-1} * \frac{4}{2n-1} * \left(\frac{1}{2^{2n-1}} + \frac{1}{3^{2n-1}} \right)$$

Să se determine valoarea lui s_n , astfel încât suma să fie îndeplinită relația $|s_n - s_{n-1}| < eps$.

6. Se cunoaște seria convergentă:

$$\sum_{n=1}^{\infty} 3 \frac{\sqrt{3}}{2} * \left(1 + \frac{1!}{6} + \frac{2!}{6 * 10} + \frac{3!}{6 * 10 * 14} + \dots + \frac{n!}{6 * 10 * 14 * \dots * (6 + 4 * (n-1))} \right)$$

Să se determine valoarea lui s_n , astfel încât suma să fie îndeplinită relația $|s_n - s_{n-1}| < eps$.

7. Se cunoaște seria convergentă:

$$\sum_{n=0}^{\infty} \left(\frac{1}{1} * \frac{1}{9^0} \right) + \left(\frac{1}{3} * \frac{1}{9^1} \right) + \left(\frac{1}{5} * \frac{1}{9^2} \right) + \left(\frac{1}{7} * \frac{1}{9^3} \right) + \dots + \left(\frac{1}{2 * n + 1} * \frac{1}{9^n} \right)$$

Să se determine valoarea lui s_n , astfel încât suma să fie îndeplinită relația $|s_n - s_{n-1}| < eps$.

8. Se cunoaște seria convergentă:

$$\sum_{n=1}^{\infty} \frac{1!}{1 * 1} + \frac{2!}{1 * 3 * 3} + \frac{3!}{1 * 3 * 5 * 7 * 7} + \frac{4!}{1 * 3 * 5 * 7 * 9 *} \dots + \frac{n!}{1 * 3 * 5 * \dots * (2n+1) * (2n+1)}$$

Să se determine valoarea lui s_n , astfel încât suma să fie îndeplinită relația $|s_n - s_{n-1}| < eps$.

- a) Să se scrie o funcție care are ca parametru un număr real ε și calculează numărul real e cu precizia ε dată. Se va aplica formula $x_n = 1 + \frac{1}{1!} + \frac{1}{2!} + \dots + \frac{1}{n!}$. Considerăm că aproximația numărului e cu precizia ε este valoarea x_{n+1} având proprietatea că $|x_{n+1} - x_n| < \varepsilon$.

```

#include <iostream>
using namespace std;
//descriere: e=xn =1+1/1!+1/2!+...+1/n! calculati valoarea lui e aproximarea
//          eps pana cand |x(n+1)-x(n)|<eps
//int: eps
//out: e cu aproximare eps

```

```
float expresie (float eps)
{float n=0;
  float p=1; // 1!
  float e=1;
  float t=1/p;
  while (t>=eps)
  {n++;
   p=p*n;
   t=1/p;
   e=e+t;
  }
  return e;
}
int main()
{float eps;
  cout<<"eps=";
  cin>>eps;
  cout<<expresie(eps);
  return 0;
}
```