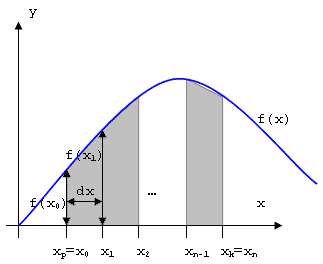
**Całkowanie numeryczne – metoda trapezów**

Załóżmy, że chcemy obliczyć całkę z funkcji *f(x)* w przedziale *<xp; xk>*. Definicja całki oznaczonej Riemana, mówi nam, że wartość całki równa jest sumie pól obszarów pod wykresem krzywej w zadanym przedziale całkowania. Sumę taką możemy obliczyć w przybliżeniu dzieląc obszar całkowania na *n* równych części. Dla każdej takiej części możemy wyznaczyć trapez, który w przybliżeniu będzie odpowiadał polu obszaru pod wykresem krzywej.



Jak już wspomnieliśmy przedział całkowania *<xp; xk>* podzielimy na *n* równych części. Szerokość każdej z nich wynosić będzie zatem:

Taka też będzie wysokość każdego z trapezów. Podstawy *i*-tego trapezu będą wynosić odpowiednio:

dla *i* = 1, 2, ..., *n*, gdzie *xi = xp + i\*dx*.

Pole *i*-tego trapezu zgodnie ze wzorem wynosić będzie:

Całkę w zadanym przedziale uzyskamy dodając do siebie pola wszystkich wyznaczonych trapezów, wynosić będzie ona zatem:

W praktyce w pętli dodajemy do siebie wszystkie wartości funkcji od 1 do n-1, a potem dwie wartości brzegowe podzielone przez dwa. Całość mnożymy przez *dx* i otrzymujemy w ten sposób wynik.

Przykładowa implementacja w języku C:

#include "stdio.h"

//funkcja dla ktorej obliczamy calke

float func(float x) {

return x\*x+3;

}

int main(){

float xp, xk, dx, calka;

int i, n;

printf("Podaj poczatek przedzialu calkowania\n");

scanf("%f", &xp);

printf("Podaj koniec przedzialu calkowania\n");

scanf("%f", &xk);

printf("Podaj dokladnosc calkowania\n");

scanf("%d", &n);

dx = (xk - xp) / (float)n;

calka = 0;

for (i=1; i<n; i++) {

calka += func(xp + i \* dx);

}

calka += (func(xp) + func(xk)) / 2;

calka \*= dx;

printf("Wartosc calki wynosi w przyblizeniu %f\n", calka);

return 0;

}