Υπολογιστικές Μέθοδοι

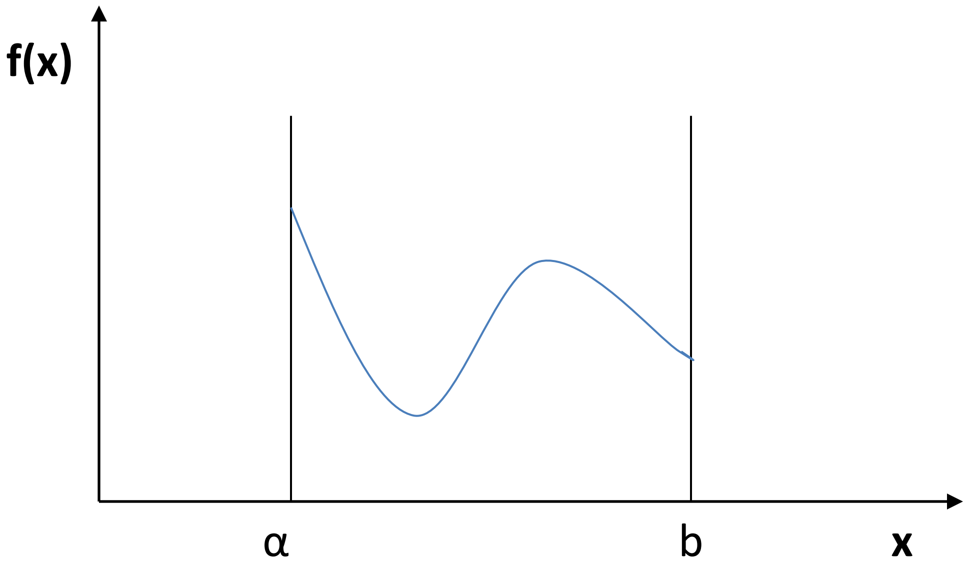
http://eclass.uoa.gr/courses/PHYS186/

Διδάσκοντες:
Φ. Διάκονος
Δ. Φασουλιώτης

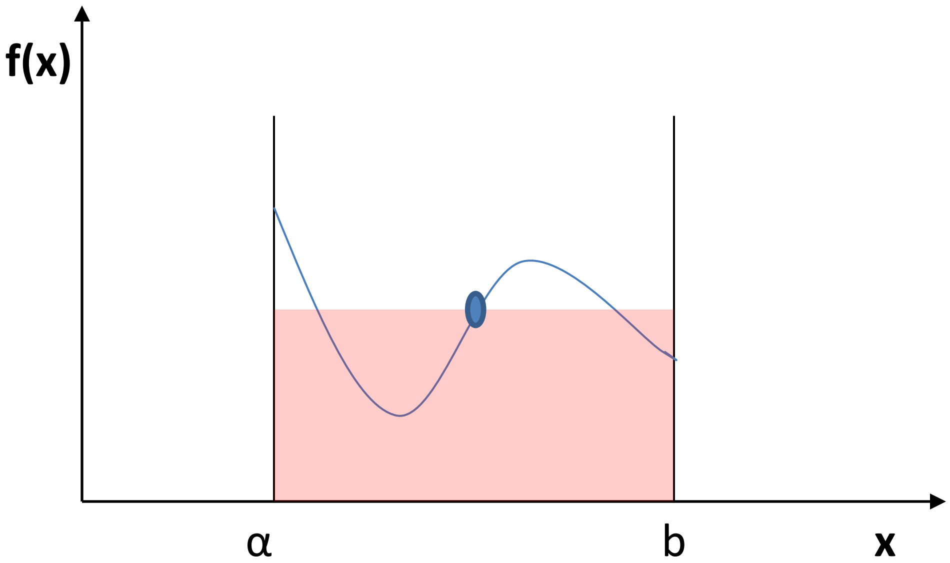
- Αριθμητική ολοκλήρωση

Αριθμητική Ολοκλήρωση: Εισαγωγή

Γενικά : $$\int_a^b f(x)dx \approx \sum_{i=1}^{n} c_i f(x_i)$$



$$\int_a^b f(x)dx \approx \sum_{i=1}^{n} c_i f(x_i)$$



Έστω δειγματοληψία ενός σημείου

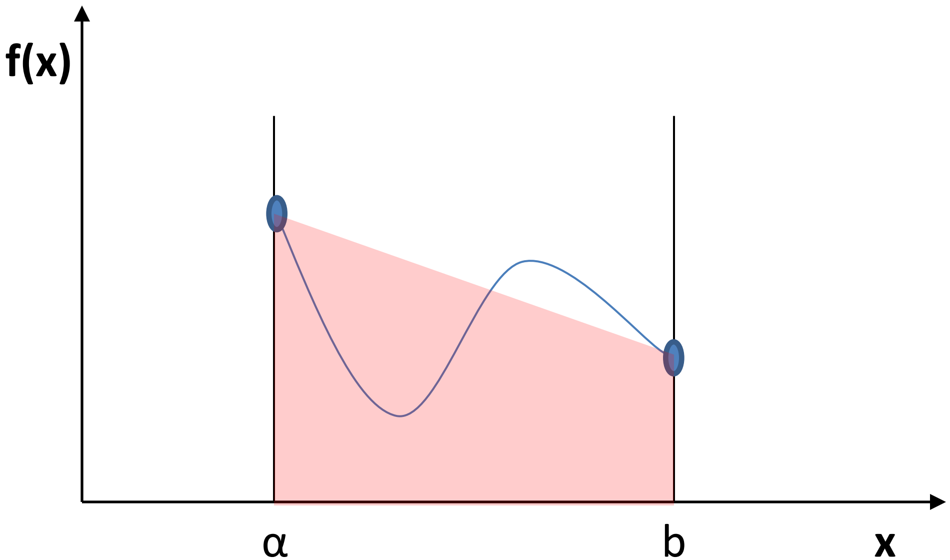
$$\int_a^b f(x)dx \approx c_1 f(a + \frac{b-a}{2})$$

$$c_1 = b - a$$

Μέθοδος ορθογωνίου

Αριθμητική Ολοκλήρωση : Εισαγωγή

$$\int_a^b f(x)dx \approx \sum_{i=1}^{n} c_i f(x_i)$$



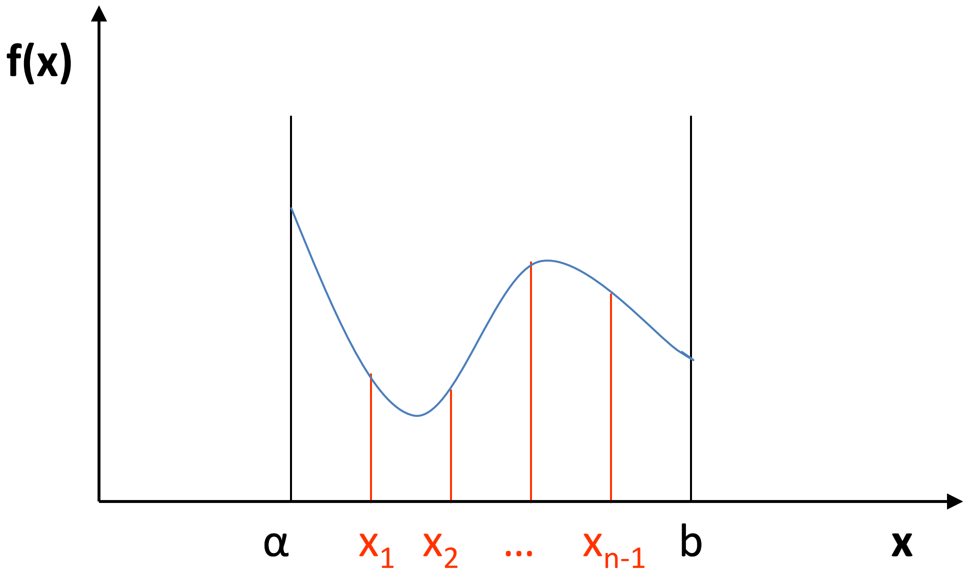
Έστω δειγματοληψία δύο σημείων

$$\int_a^b f(x)dx \approx c_1 f(a) + c_2 f(b)$$

$$c_1 = c_2 = \frac{b-a}{2}$$

Μέθοδος τραπεζίου

Γενικά : $$\int_a^b f(x)dx \approx \sum_{i=1}^{n} c_i f(x_i)$$



Στη γενική περίπτωση πρέπει να χρησιμοποιήσουμε ένα σχήμα παρεμβολής στην συνάρτηση *f* ώστε να μπορέσουμε να υπολογίσουμε τους συντελεστές c_i

Πολυώνυμα παρεμβολής Lagrange και ισαπέχοντα σημεία δειγματοληψίας

$$P_n(x) = \sum_{i=0}^{n} f(x_i) L_{n,i}(x) \qquad L_{n,i}(x) = \prod_{k=0, i\neq k}^{n} \frac{(x - x_k)}{(x_i - x_k)}$$

Οπότε:

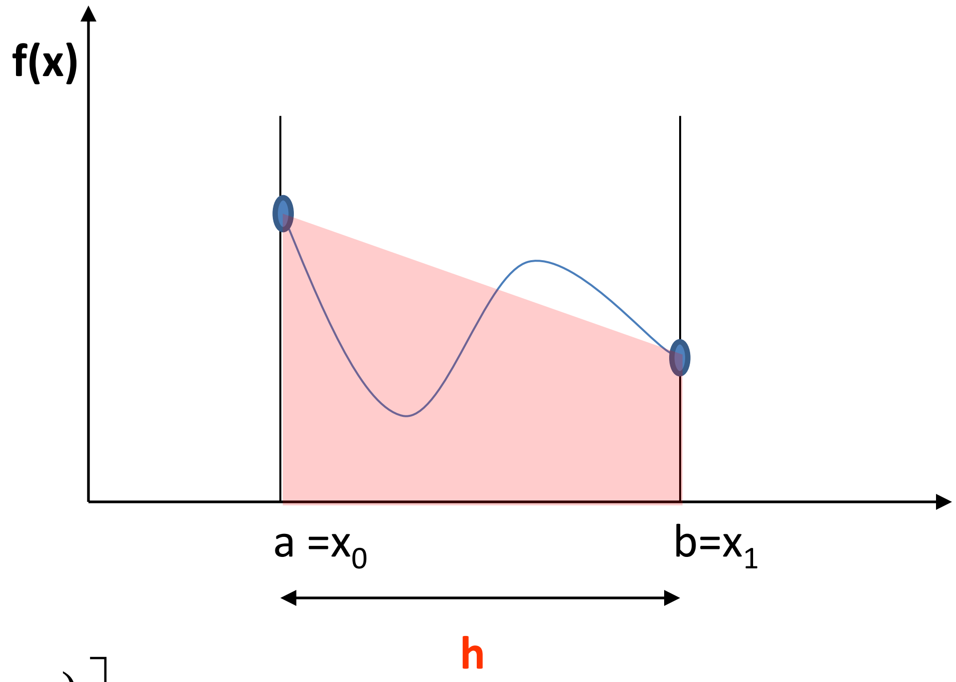
$$\int_a^b f(x)dx \approx \int_a^b P_n(x)dx \quad = \sum_{i=0}^{n} \int_a^b f(x_i) L_{n,i}(x) dx$$

$$= \sum_{i=0}^{n} a_i f(x_i)$$

με: $$a_i = \int_a^b L_{n,i}(x)dx$$

Πολυώνυμο παρεμβολής Lagrange P_1

$$P_1(x) = f(x_0)\frac{(x-x_1)}{(x_0-x_1)} + f(x_1)\frac{(x-x_0)}{(x_1-x_0)}$$



Οπότε:

$$\int_a^b f(x)dx \approx \int_{a\equiv x_0}^{b\equiv x_1}\left[f(x_0)\frac{(x-x_1)}{(x_0-x_1)} + f(x_1)\frac{(x-x_0)}{(x_1-x_0)}\right]dx$$

$$= \frac{f(x_0)}{(x_1-x_0)}\int_{x_0}^{x_1}(x_1-x)dx + \frac{f(x_1)}{(x_1-x_0)}\int_{x_0}^{x_1}(x-x_0)dx +$$

$$= \frac{1}{2}(x_1-x_0)\left[f(x_0)+f(x_1)\right]$$

$$\equiv \frac{h}{2}\left[f(x_0)+f(x_1)\right]$$

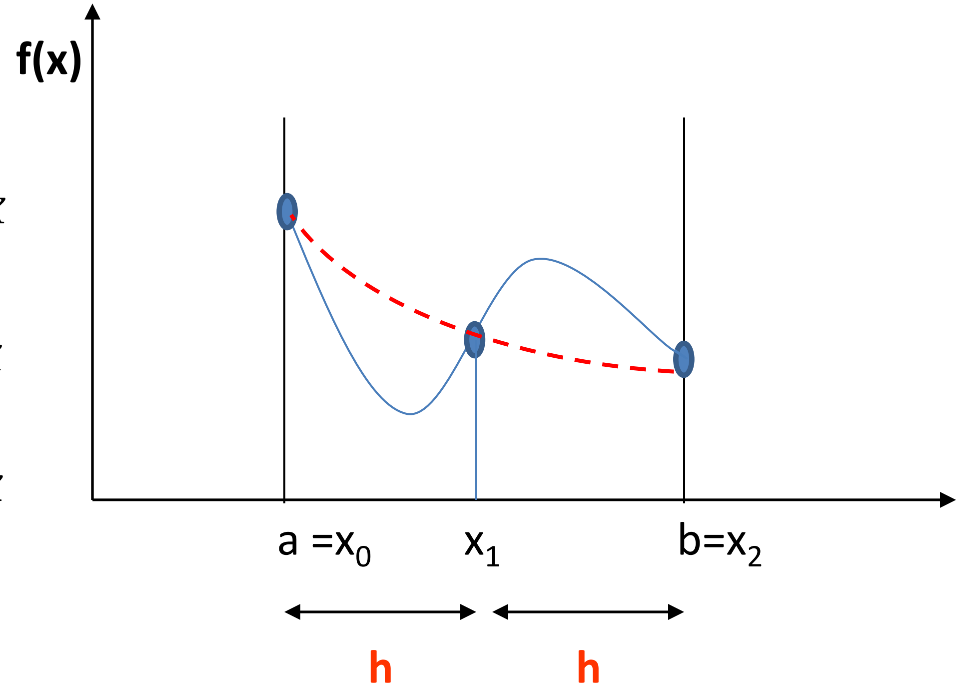
$\boxed{\text{Μέθοδος Τραπεζίου}}$

Πολυώνυμο παρεμβολής Lagrange P_2

$$P_2(x) = f(x_0)\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + f(x_1)\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + f(x_2)\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$

Οπότε:

$$\begin{aligned}\int_a^b f(x)dx &\approx \frac{f(x_0)}{2h^2}\int_{x_0}^{x_2}(x-x_1)(x-x_2)dx \\ &-\frac{f(x_1)}{h^2}\int_{x_0}^{x_2}(x-x_0)(x-x_2)dx \\ &+\frac{f(x_2)}{2h^2}\int_{x_0}^{x_2}(x-x_0)(x-x_1)dx \\ &=\frac{h}{3}\left[f(x_0)+4f(x_1)+f(x_2)\right]\end{aligned}$$



Μέθοδος Simpson

Σφάλμα αποκοπής προκύπτει από την ολοκλήρωση του σφάλματος του πολυωνύμου παρεμβολής

Για τη μέθοδο τραπεζίου (παρεμβολή 1ης τάξης):

$$f(x) = P_1(x) + {f''(\xi)}\big/{2}\,(x - x_0)(x - x_1)$$

$$\begin{aligned}
E(\tau\rho\alpha\pi) &\approx {f''(\xi)}\big/{2} \int_{x_0}^{x_1} (x - x_0)(x - x_1)dx \\
&= {f''(\xi)}\big/{2} \int_0^h x(x-h)dx \\
&= {f''(\xi)}\big/{2} \left(\frac{1}{3}x^3 \Big|_0^h - \frac{hx^2}{2} \Big|_0^h \right) \\
& {f''(\xi)}\big/{2} \left(\frac{h^3}{3} - \frac{h^3}{2} \right)
\end{aligned}$$

$$\boxed{E(\tau\rho\alpha\pi) = -{h^3}\big/{12}\, f''(\xi)}$$

Σφάλμα αποκοπής προκύπτει από την ολοκλήρωση του σφάλματος του πολυωνύμου παρεμβολής

Για τη μέθοδο Simpson (παρεμβολή 2ης τάξης):

$$f(x) = P_2(x) + {f^{(3)}(\xi)}\big/{6}\,(x-x_0)(x-x_1)(x-x_2)$$

$$\begin{aligned}
E(Simpson) &\approx {f^{(3)}(\xi)}\big/{6} \int_{x_0}^{x_2} (x-x_0)(x-x_1)(x-x_2)dx \\
&= {f^{(3)}(\xi)}\big/{6} \int_{-h}^{h} (x+h)x(x-h)dx \\
&= {f^{(3)}(\xi)}\big/{6} \left(\int_{-h}^{h} (x^3 - hx^2 + hx^2 - h^2x)dx \right) \\
& {f^{(3)}(\xi)}\big/{6} \left(\frac{1}{4}x^4 \Big|_{-h}^{h} - \frac{h^2x^2}{2} \Big|_{-h}^{h} \right) \\
&= 0!
\end{aligned}$$

$$\boxed{E(Simpson) \approx -{h^5}\big/{90}\, f^{(4)}(\xi)}$$

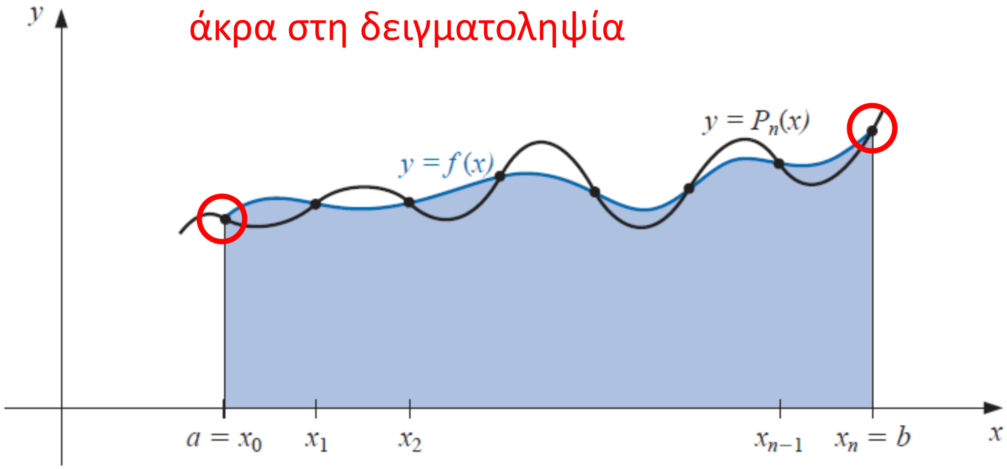
Αριθμητική Ολοκλήρωση: Κλειστές μέθοδοι Newton-Cotes

$$\int_a^b f(x)dx \approx \int_a^b P_n(x)dx$$

$$= \sum_{i=0}^{n} \int_a^b f(x_i) L_{n,i}(x)dx = \sum_{i=0}^{n} a_i f(x_i)$$

$$a_i = \int_a^b L_{n,i}(x)dx$$

Κλειστές, διότι περιλαμβάνουν τα άκρα στη δειγματοληψία



Για n άρτιο

$$\int_a^b f(x)\,dx = \sum_{i=0}^{n} a_i f(x_i) + \frac{h^{n+3} f^{(n+2)}(\xi)}{(n+2)!} \int_0^n t^2(t-1)\cdots(t-n)\,dt,$$

Για n περιττό

$$\int_a^b f(x)\,dx = \sum_{i=0}^{n} a_i f(x_i) + \frac{h^{n+2} f^{(n+1)}(\xi)}{(n+1)!} \int_0^n t(t-1)\cdots(t-n)\,dt$$

$n = 1$: Trapezoidal rule

$$\int_{x_0}^{x_1} f(x)\,dx = \frac{h}{2}[f(x_0) + f(x_1)] - \frac{h^3}{12} f''(\xi), \quad \text{where} \quad x_0 < \xi < x_1.$$

$n = 2$: Simpson's rule

$$\int_{x_0}^{x_2} f(x)\,dx = \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)] - \frac{h^5}{90} f^{(4)}(\xi), \quad \text{where} \quad x_0 < \xi < x_2.$$

$n = 3$: Simpson's Three-Eighths rule

$$\int_{x_0}^{x_3} f(x)\,dx = \frac{3h}{8}[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)] - \frac{3h^5}{80} f^{(4)}(\xi),$$

$$\text{where} \quad x_0 < \xi < x_3.$$

$n = 4$:

$$\int_{x_0}^{x_4} f(x)\,dx = \frac{2h}{45}[7f(x_0) + 32f(x_1) + 12f(x_2) + 32f(x_3) + 7f(x_4)] - \frac{8h^7}{945} f^{(6)}(\xi),$$

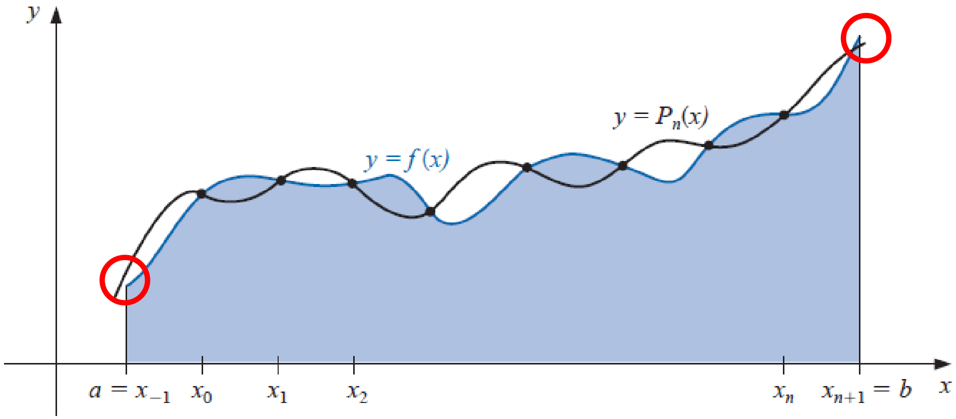
$$\text{where} \quad x_0 < \xi < x_4.$$

$$\int_a^b f(x)dx \approx \int_a^b P_n(x)dx$$

$$= \sum_{i=0}^{n} \int_a^b f(x_i) L_{n,i}(x)dx = \sum_{i=0}^{n} a_i f(x_i)$$

$$a_i = \int_a^b L_{n,i}(x)dx$$

Ανοιχτές, διότι ΔΕΝ περιλαμβάνουν τα άκρα στη δειγματοληψία



Για n άρτιο

$$\int_a^b f(x)\,dx = \sum_{i=0}^{n} a_i f(x_i) + \frac{h^{n+3} f^{(n+2)}(\xi)}{(n+2)!} \int_{-1}^{n+1} t^2(t-1)\cdots(t-n)\,dt,$$

Για n περιττό

$$\int_a^b f(x)\,dx = \sum_{i=0}^{n} a_i f(x_i) + \frac{h^{n+2} f^{(n+1)}(\xi)}{(n+1)!} \int_{-1}^{n+1} t(t-1)\cdots(t-n)\,dt$$

$n = 0$: Midpoint rule

$$\int_{x_{-1}}^{x_1} f(x)\,dx = 2hf(x_0) + \frac{h^3}{3}f''(\xi), \quad \text{where} \quad x_{-1} < \xi < x_1.$$

$n = 1$:

$$\int_{x_{-1}}^{x_2} f(x)\,dx = \frac{3h}{2}[f(x_0) + f(x_1)] + \frac{3h^3}{4}f''(\xi), \quad \text{where} \quad x_{-1} < \xi < x_2.$$

$n = 2$:

$$\int_{x_{-1}}^{x_3} f(x)\,dx = \frac{4h}{3}[2f(x_0) - f(x_1) + 2f(x_2)] + \frac{14h^5}{45}f^{(4)}(\xi),$$

$$\text{where} \quad x_{-1} < \xi < x_3.$$

$n = 3$:

$$\int_{x_{-1}}^{x_4} f(x)\,dx = \frac{5h}{24}[11f(x_0) + f(x_1) + f(x_2) + 11f(x_3)] + \frac{95}{144}h^5 f^{(4)}(\xi),$$

$$\text{where} \quad x_{-1} < \xi < x_4.$$

Κανόνας τραπεζίου

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
using namespace std;
double myfunc(double );
double myfunc(double x)
{
  double value=exp(x);
  return value;
}
void trapezio(double a, double b)
{
  double h=b-a;
  double inte=(h/2)*(myfunc(a)+myfunc(b));
  cout << inte <<endl;
}
```

Κανόνας Simpson 1/3

```
void simpson13(double a, double b)
{
  double h=(b-a)/2;
  double inte=(h/3)*(myfunc(a)+4*myfunc(a+h)+myfunc(b));
}
```

Κανόνας Simpson 3/8

```
void simpson38(double a, double b)
{
  double h=(b-a)/3;
  double inte=(3*h/8)*(myfunc(a)+3*myfunc(a+h)+3*myfunc(a+2*h)+myfunc(b));
}
```

Παράδειγμα: **Μέθοδος Simpson** για την εκτίμηση του $I = \int_0^4 e^x dx$

$$I = \int_0^4 e^x dx \approx \frac{2}{3}\left(e^0 + 4e^2 + e^4\right) = 56.76958$$

Απόκλιση -3.17143

Παράδειγμα: Μέθοδος Simpson για την εκτίμηση του $I = \int_0^4 e^x dx$

$$I = \int_0^4 e^x dx \approx \frac{2}{3}\left(e^0 + 4e^2 + e^4\right) = 56.76958$$

Απόκλιση -3.17143

Αν ξανακάνουμε τον υπολογισμό χωρίζοντας το αρχικό διάστημα σε 2 υποδιαστήματα:

$$I_2 = \int_0^4 e^x dx \quad = \int_0^2 e^x dx + \int_2^4 e^x dx$$

$$\approx \frac{1}{3}\left(e^0 + 4e \quad + e^2\right) + \frac{1}{3}\left(e^2 + 4e^3 + e^4\right) = 53.86385$$

Απόκλιση -0.26570

Παράδειγμα: **Μέθοδος Simpson** για την εκτίμηση του $I=\int_0^4 e^x dx$

$$I=\int_0^4 e^x dx \approx \frac{2}{3}\left(e^0+4e^2+e^4\right)=56.76958$$

Απόκλιση -3.17143

Αν ξανακάνουμε τον υπολογισμό χωρίζοντας το αρχικό διάστημα σε 2 υποδιαστήματα:

$$I_2=\int_0^4 e^x dx \quad =\int_0^2 e^x dx+\int_2^4 e^x dx$$

$$\approx \frac{1}{3}\left(e^0+4e\ \ +e^2\right)+\frac{1}{3}\left(e^2+4e^3+e^4\right)=53.86385$$

Απόκλιση -0.26570

Αν ξανακάνουμε τον υπολογισμό χωρίζοντας το αρχικό διάστημα σε 4 υποδιαστήματα:

$$I_4=\int_0^4 e^x dx \quad =\int_0^1 e^x dx+\int_1^2 e^x dx+\int_2^3 e^x dx+\int_3^4 e^x dx$$

$$\approx \frac{1}{6}\left(e^0+4e^{0.5}+e^1\right)+\frac{1}{6}\left(e^1+4e^{1.5}+e^2\right)$$

$$+\frac{1}{6}\left(e^2+4e^{2.5}+e^3\right)+\frac{1}{6}\left(e^3+4e^{3.5}+e^4\right)=53.61622$$

Απόκλιση -0.01807

Το σφάλμα της μεθόδου τραπεζίου σε ένα διάστημα είναι:

$$E(\tau\rho\alpha\pi) \approx -{h^3}/{12}\, f''(\xi)$$,όπου *h=b-a*

Αν χωρίσουμε το διάστημα ολοκλήρωσης σε n διαστήματα, τότε *h=(b-a)/n,* και το συνολικό σφάλμα θα είναι n φορές αυτό του κάθε υποδιαστήματος:

$$E(\tau\rho\alpha\pi) \approx -\frac{(b-a)}{12} h^2 f''(\xi) = -\frac{(b-a)^3}{12}\frac{1}{n^2} f''(\xi)$$

Το σφάλμα της μεθόδου Simpson σε ένα διάστημα είναι:

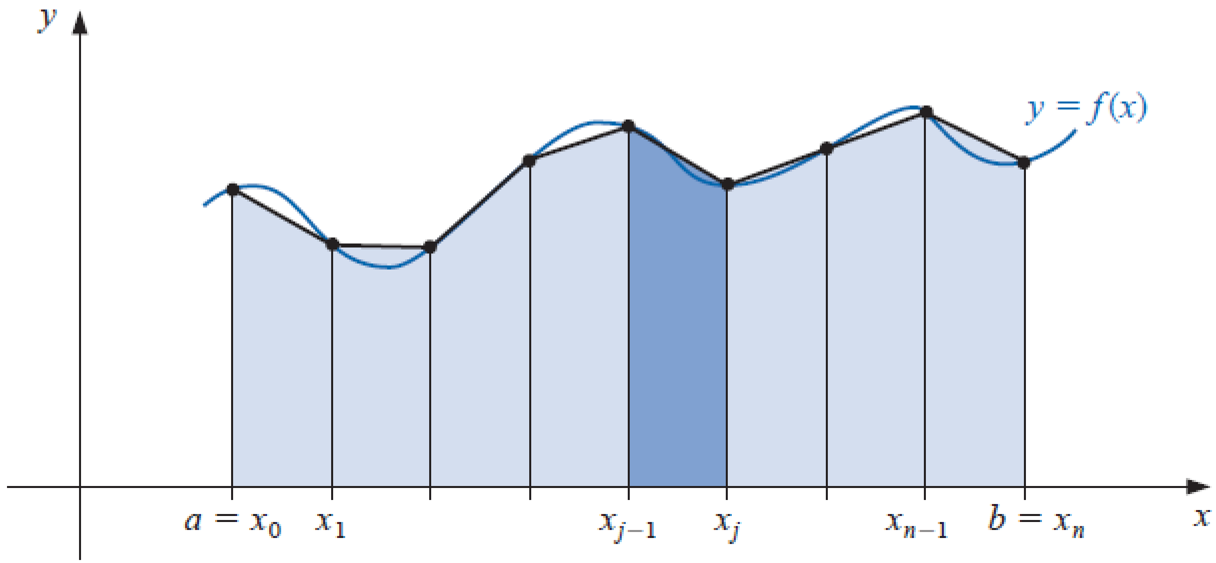
$$E(Simpson) \approx -{h^5}/{90}\, f^{(4)}(\xi)$$,όπου *h=(b-a)/2*

Αν χωρίσουμε το διάστημα ολοκλήρωσης σε n διαστήματα, τότε *h=(b-a)/2n*, και το συνολικό σφάλμα θα είναι n φορές αυτό του κάθε υποδιαστήματος:

$$E(Simpson) \approx -\frac{(b-a)}{180} h^4 f^{(4)}(\xi) = \frac{(b-a)^5}{180}\frac{1}{16n^4} f^{(4)}(\xi)$$

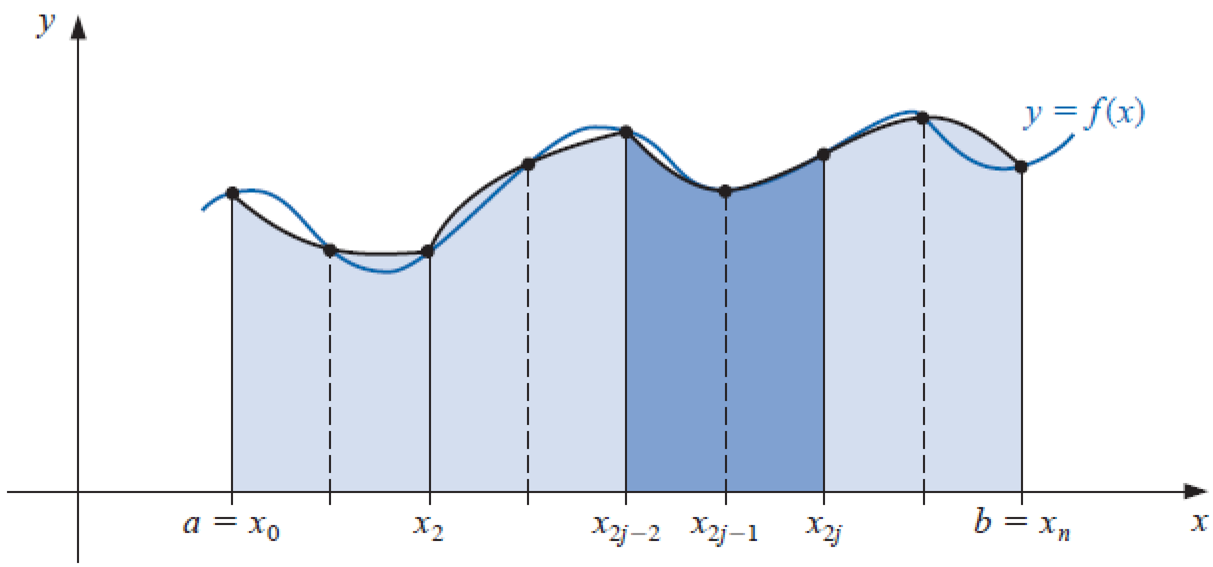
Γενικευμένος κανόνας Τραπεζίου

$$\int_a^b f(x)dx = \frac{h}{2}\left[f(a) + 2\sum_{i=1}^{n-1} f(x_i) + f(b) \right] - \frac{(b-a)}{12} h^2 f''(\xi)$$



Γενικευμένος κανόνας Simpson, όπου m=2n

$$\int_a^b f(x)dx = \frac{h}{3}\left[f(a) + 2\sum_{i=1}^{(m/2)-1} f(x_{2i}) + 4\sum_{i=1}^{(m/2)} f(x_{2i-1}) + f(b) \right] - \frac{(b-a)}{180} h^4 f^{(4)}(\xi)$$



Αριθμητική Ολοκλήρωση: Μέθοδοι Newton-Cotes –Ολοκλήρωση κατά τμήματα

$$\int_a^b f(x)dx = \frac{h}{2}\left[f(a) + 2\sum_{i=1}^{n-1} f(x_i) + f(b) \right] - \frac{(b-a)}{12} h^2 f''(\xi)$$

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
double myfunc(double x)
{
    double ff=0.;
    ff= exp(x); // ή ότιδήποτε άλλο
    return ff;
}
void trapezio(double a, double b, int n)
{
    double h=(b-a)/n;
    double integral=0.5*h*(myfunc(a)+myfunc(b));
    for (int i=1; i<n; i++){
         double xx=a+h*i;
         integral+=h*myfunc(xx);
        }
     cout<< n <<" "<< integral<< endl;
}
```

$$\int_a^b f(x)dx = \frac{h}{3}\left[f(a) + 2\sum_{i=1}^{(m/2)-1} f(x_{2i}) + 4\sum_{i=1}^{(m/2)} f(x_{2i-1}) + f(b) \right] - \frac{(b-a)}{180} h^4 f^{(4)}(\xi)$$

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
double myfunc(double x)
{
    double ff=0.;
    ff= exp(x); // ή οτιδήποτε άλλο
    return ff;
}
void simpson(double a, double b, int n)
{
    double h=(b-a)/2/n;
    double integral=h*(myfunc(a)+myfunc(b))/3;
    for (int i=1; i<2*n; i++){
       double xx=a+h*i;
       if(i%2==0){integral+=2*h*myfunc(xx)/3;}
       else {integral+=4*h*myfunc(xx)/3;}
     }
     cout<< n <<" "<< integral<< endl;
}
```

Έστω ότι υπολογίζουμε το ολοκλήρωμα με τη μέθοδο **Simpson** σε n διαστήματα με τιμή S_n και η απόκλιση από την πραγματική τιμή είναι E_n.

$$I = \int_a^b f(x)dx = S_n + E_n$$

Υπολογίζουμε το ίδιο ολοκλήρωμα με τη μέθοδο Simpson σε 2n διαστήματα με τιμή S_{2n} και η αντίστοιχη απόκλιση είναι E_{2n}.

$$I = \int_a^b f(x)dx = S_{2n} + E_{2n}$$

Δεδομένου ότι :

$$E(Simpson) \approx \frac{(b-a)^5}{180}\frac{1}{16n^4} f^{(4)}(\xi) \Rightarrow E_{2n} = \frac{1}{16} E_n$$

Αφαιρώντας κατά μέλη τις δύο παραπάνω σχέσεις και αντικαθιστώντας το E_n, προκύπτει

$$S_{2n} - S_n = E_n - E_{2n} = 16E_{2n} - E_{2n} = 15E_{2n} \Rightarrow$$

$$\boxed{E_{2n} = \frac{S_{2n} - S_n}{15}}$$

$$I = \int_0^4 e^x dx$$

```
void simpson(double a, double b, int n)
{
// υπολογισμός ολοκληρώματος
double h=(b-a)/2/n;
double integral_b=0, integral_c=0;
double integral_a=(myfunc(a)+myfunc(b));
for (int i=1; i<2*n; i++){
         double xx=a+h*i;
         if(i%2==0){integral_b+=myfunc(xx);}
         else {integral_c+=myfunc(xx);}
         }
double simpson_n=(h/3.)*(integral_a+2*integral_b+4*integral_c);
// υπολογισμός σφάλματος
int m=n/2;
h=(b-a)/2/m;
integral_b=0, integral_c=0;
for (int i=1; i<2*m; i++){
         double xx=a+h*i;
         if(i%2==0){integral_b+=myfunc(xx);}
         else {integral_c+=myfunc(xx);}
         }
double simpson_m=(h/3.)*(integral_a+2*integral_b+4*integral_c);
double simpson_error=fabs(simpson_n-simpson_m)/15.;
//εκτύπωση αποτελεσμάτων
double trueval= exp(4.)-exp(0.);
cout<< n <<" "<< simpson_n<< " +- "<< simpson_error;
cout<<" apoklisi "<<simpson_n-trueval <<endl;
}
```

Υπάρχουν όμως και τα σφάλματα στρογγυλοποίησης

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n} a_i f(x_i)$$

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n} a_i \hat{f}(x_i) + \sum_{i=0}^{n} a_i \varepsilon_i$$ με $$\varepsilon_i \le \varepsilon$$

$$\int_a^b f(x)dx - \sum_{i=0}^{n} a_i \hat{f}(x_i) \le +\varepsilon \sum_{i=0}^{n} a_i$$

Υπάρχουν όμως και τα σφάλματα στρογγυλοποίησης

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n} a_i f(x_i)$$

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n} a_i \hat{f}(x_i) + \sum_{i=0}^{n} a_i \varepsilon_i \quad \text{με} \quad \varepsilon_i \leq \varepsilon$$

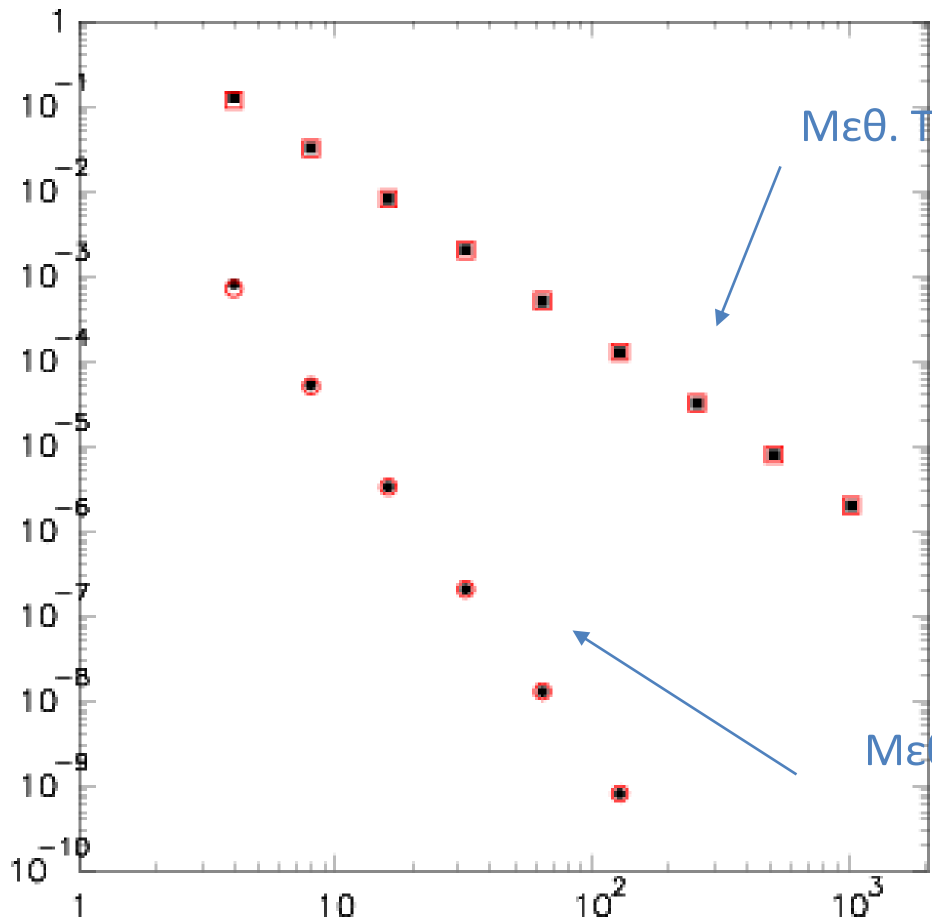
$$\int_a^b f(x)dx - \sum_{i=0}^{n} a_i \hat{f}(x_i) \leq +\varepsilon \sum_{i=0}^{n} a_i \Longrightarrow \textbf{(b-α)}$$

Σφάλμα στρογγυλοποίησης **≤ ε (b-α)**

Ποσοστιαίο σφάλμα

Πραγματική απόκλιση

Εκτιμούμενο σφάλμα



Μεθ. Τραπεζίου

Μεθ. Simpson

Αριθμός Υποδιαστημάτων

Αριθμητική Ολοκλήρωση: Μέθοδος Romberg

Υπολογίζουμε ένα ολοκλήρωμα με τη μέθοδο τραπεζίου σε 1 και 2 διαστήματα

$$T_2 = \frac{h}{2}\left[f(x_0) + 2f(x_1) + f(x_2)\right] \quad -\frac{(b-a)}{12}h^2 f''(\xi)$$

$$T_1 = \frac{2h}{2}\left[f(x_0) + f(x_2)\right] \quad -\frac{(b-a)}{12}(2h)^2 f''(\xi)$$

Από τις παραπάνω εκφράσεις διαπιστώνουμε ότι για να εξαφανίσουμε την απόκλιση που είναι ανάλογη της *f''(ξ)* αρκεί να πάρουμε τον γραμμικό συνδυασμό των παραπάνω αποτελεσμάτων $4/3T_2$-$1/3T_{1.}$ Πράγματι:

$$\begin{aligned}\frac{4}{3}T_2 - \frac{1}{3}T_1 &= \frac{4h}{6}\left[f(x_0) + 2f(x_1) + f(x_2)\right] - \frac{2h}{6}\left[f(x_0) + f(x_2)\right] \\ &= \frac{h}{3}\left[f(x_0) + 4f(x_1) + f(x_2)\right]\end{aligned}$$

…που καταλήγει στη μέθοδο Simpson

Αριθμητική Ολοκλήρωση: Μέθοδος Romberg

Διαστήματα Μεθόδου τραπεζίου					
1	$R_{1,1}$				
2	$R_{2,1}$				
4	$R_{3,1}$				
8	$R_{4,1}$				
…	…				
2^{k-1}	$R_{k,1}$				

Εφόσον στα $R_{k,1}$ εφαρμόζεται η γενικευμένη μέθοδος τραπεζίου, η ακρίβεια θα είναι ≈ $\boldsymbol{h^2}$

Αριθμητική Ολοκλήρωση: Μέθοδος Romberg

Διαστήματα Μεθόδου τραπεζίου					
1	$R_{1,1}$				
2	$R_{2,1}$	$\mathbf{R_{2,2}}$			
4	$R_{3,1}$	$\mathbf{R_{3,2}}$			
8	$R_{4,1}$	$\mathbf{R_{4,2}}$			
…	…	**…**			
2^{n-1}	$R_{k,1}$	$\mathbf{R_{k,2}}$			

$$R_{k,2} = R_{k,1} + \frac{1}{3}\left(R_{k,1} - R_{k-1,1}\right)$$

Στα $R_{k,2}$ επιτυγχάνουμε ακρίβεια ≈ $\boldsymbol{h^4}$

Διαστήματα Μεθόδου τραπεζίου					
1	$R_{1,1}$				
2	$R_{2,1}$	$R_{2,2}$			
4	$R_{3,1}$	$R_{3,2}$	$\mathbf{R_{3,3}}$		
8	$R_{4,1}$	$R_{4,2}$	$\mathbf{R_{4,3}}$		
…	…	…	**…**		
2^{n-1}	$R_{k,1}$	$R_{k,2}$	$\mathbf{R_{k,3}}$		

$$R_{k,3} = R_{k,2} + \frac{1}{15}\left(R_{k,2} - R_{k-1,2}\right)$$

Στα $R_{k,3}$ επιτυγχάνουμε ακρίβεια ≈ $\boldsymbol{h^6}$

Αριθμητική Ολοκλήρωση: Μέθοδος Romberg

Διαστήματα Μεθόδου τραπεζίου	$\sim h^2$	$\sim h^4$	$\sim h^6$	$\sim h^8$		$\sim h^{2k}$
1	$R_{1,1}$					
2	$R_{2,1}$	$R_{2,2}$				
4	$R_{3,1}$	$R_{3,2}$	$R_{3,3}$			
8	$R_{4,1}$	$R_{4,2}$	$R_{4,3}$	$\mathbf{R_{4,4}}$		
…	…	…	…	**…**	**…**	**…**
2^{n-1}	$R_{k,1}$	$R_{k,2}$	$R_{k,3}$	$\mathbf{R_{k,4}}$	**…**	$\mathbf{R_{k,k}}$

Στη γενική περίπτωση:

$$R_{k,j} = R_{k,j-1} + \frac{1}{4^{j-1} - 1}\left(R_{k,j-1} - R_{k-1,j-1}\right)$$

Αριθμητική Ολοκλήρωση: Μέθοδος Romberg

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
double myfunc(double x)
{        double ff=0.;
         ff= sin(x);
         return ff;
}
void romberg(double a, double b, double eps)
{
    double R[50][50];
    double h=(b-a);
    R[1][1]=0.5*h*(myfunc(a)+myfunc(b));
    printf("%2.12f \n", R[1][1]);
    for (int i=2; i<=50; i++){
        R[i][1]=0.5*R[i-1][1];
        for(int k=1; k<=pow(2,i-2); k++){
           R[i][1]+=0.5*h*myfunc(a+(k-0.5)*h);
         }
        printf("%2.12f ", R[i][1]);
        for(int j=2; j<=i; j++){
            R[i][j]=R[i][j-1]+(1./(pow(4,j-1)-1))*(R[i][j-1]-R[i-1][j-1]);
            printf("%2.12f ", R[i][j]);
           }
        printf("\n");
        h=0.5*h;
        if(fabs((R[i][i]-R[i-1][i-1])/R[i][i])<eps)break;
      }
}
```

$$\boxed{R_{k,1} = \frac{1}{2}\left[R_{k-1,1} + h_{k-1}\sum_{i=1}^{2^{k-2}} f(a+(2i-1)h_k)\right]}$$

$$\boxed{R_{k,j} = R_{k,j-1} + \frac{1}{4^{j-1}-1}\left(R_{k,j-1} - R_{k-1,j-1}\right)}$$

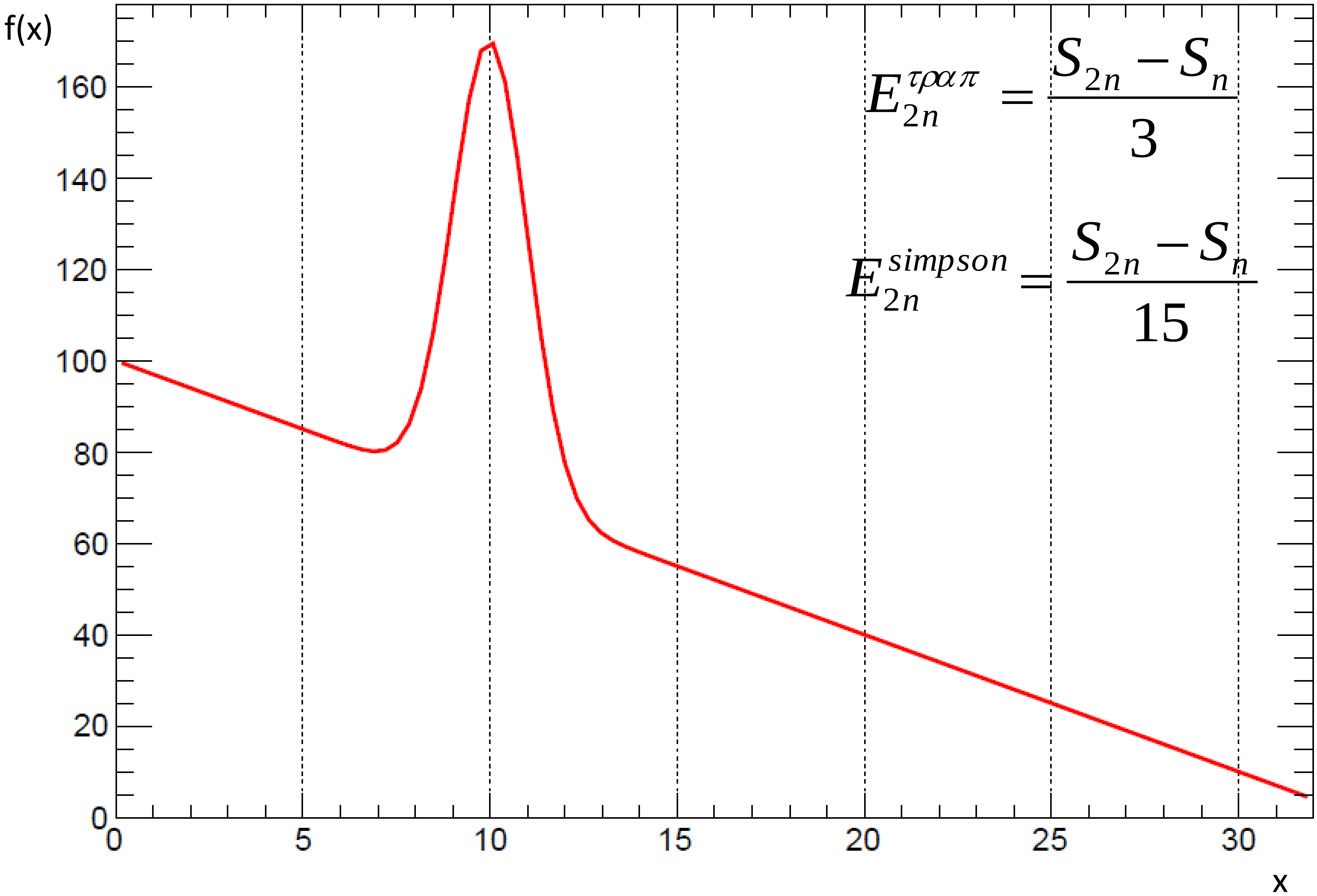
Αριθμητική Ολοκλήρωση: Μέθοδος Romberg – Παράδειγμα

Εφαρμογή της μεθόδου Romberg για τον υπολογισμό του $\int_0^{\pi} \sin(x)dx$ χρησιμοποιώντας 1, 2, 4 , 8 και 16 διαστήματα

Αποτέλεσμα

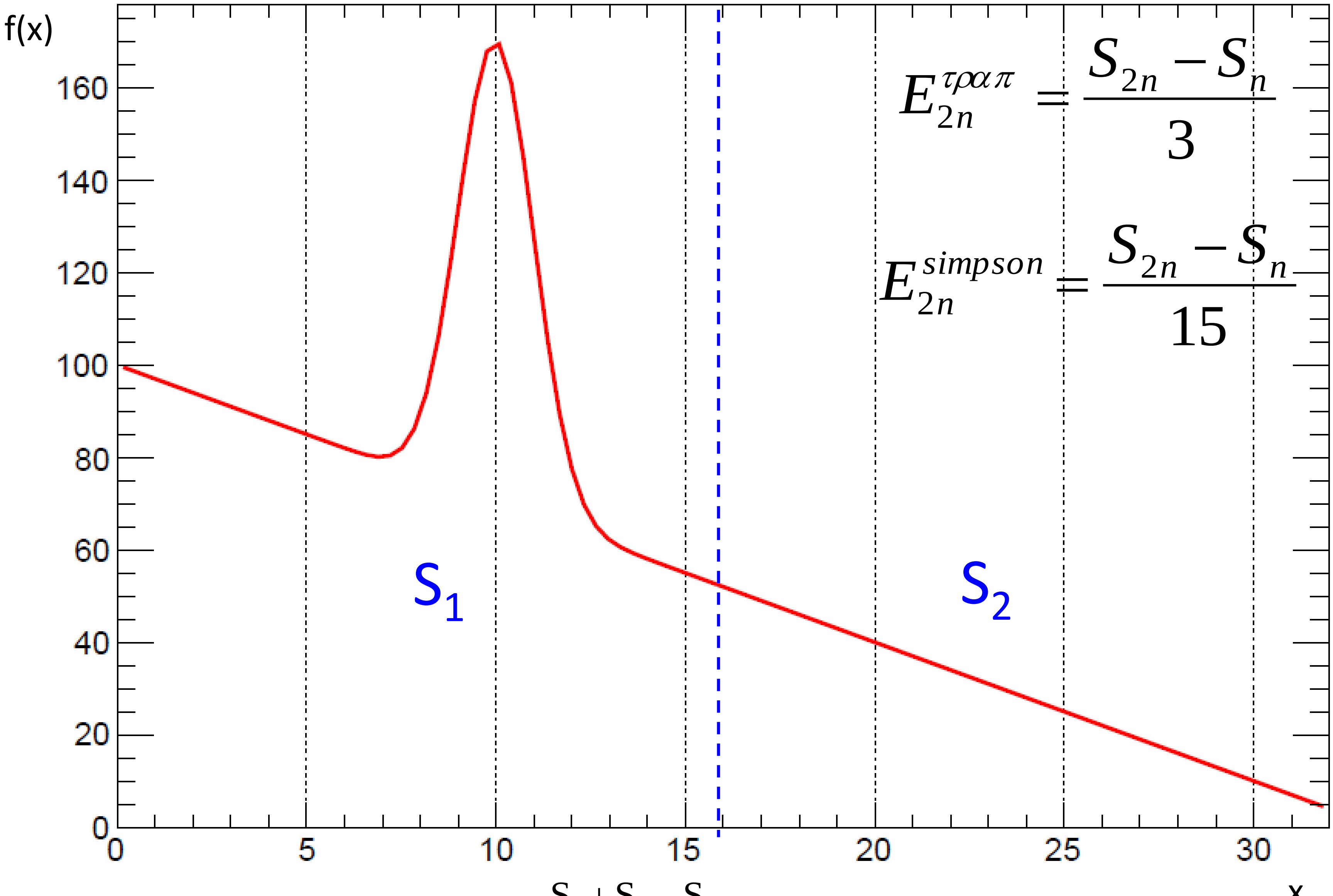
0				
1.57079633	2.09439511			
1.89611890	2.00455976	1.99857073		
1.97423160	2.00026917	1.99998313	2.00000555	
1.99357034	2.00001659	1.99999975	2.00000001	1.99999999

Αριθμητική Ολοκλήρωση: Προσαρμοζόμενες μέθοδοι



$$E_{2n}^{\tau\rho\alpha\pi} = \frac{S_{2n} - S_n}{3}$$

$$E_{2n}^{simpson} = \frac{S_{2n} - S_n}{15}$$

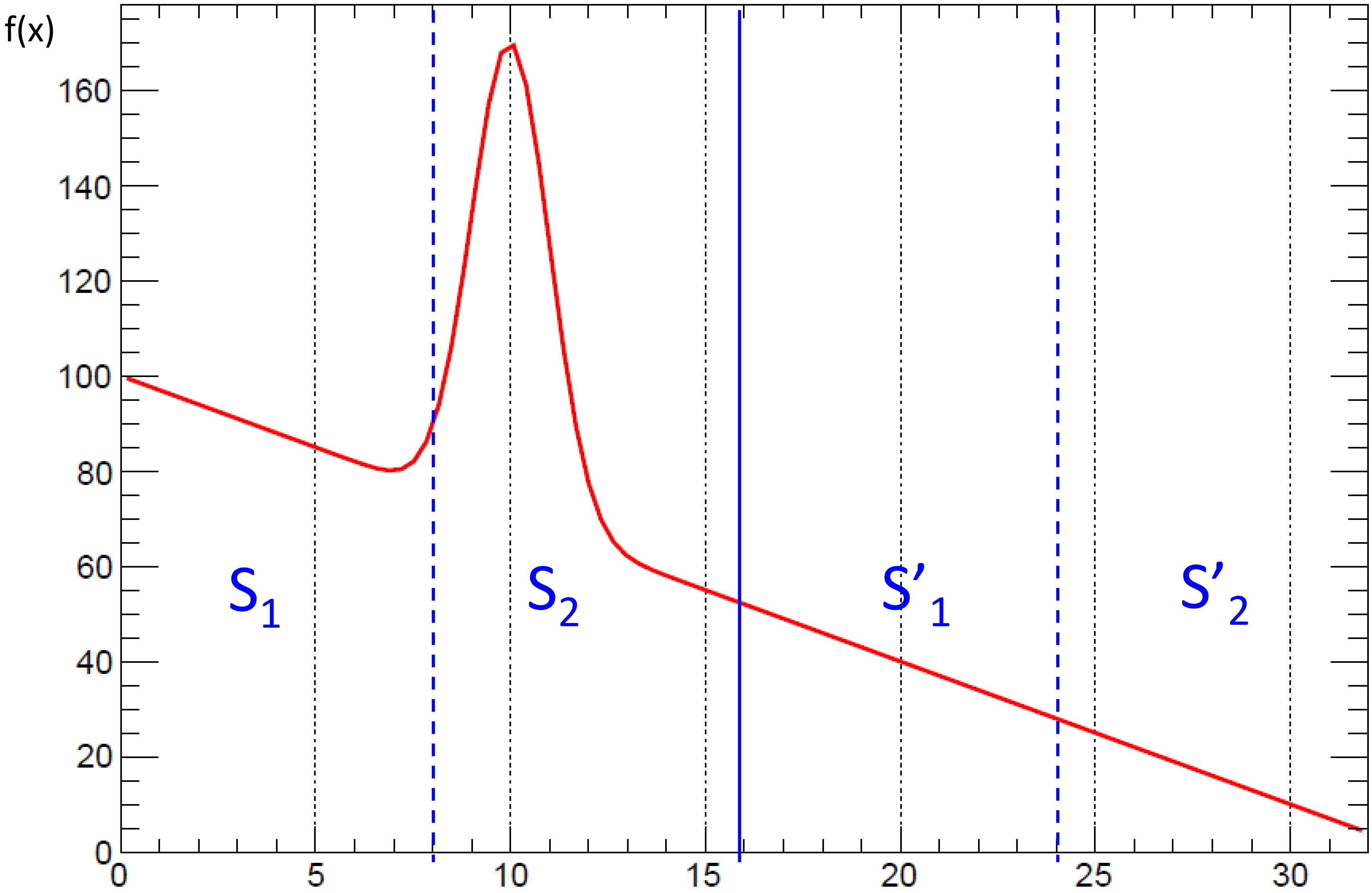


$$E_{2n}^{\tau\rho\alpha\pi} = \frac{S_{2n} - S_n}{3}$$

$$E_{2n}^{simpson} = \frac{S_{2n} - S_n}{15}$$

$$\frac{S_1 + S_2 - S_0}{15} < (?)\text{eps}$$

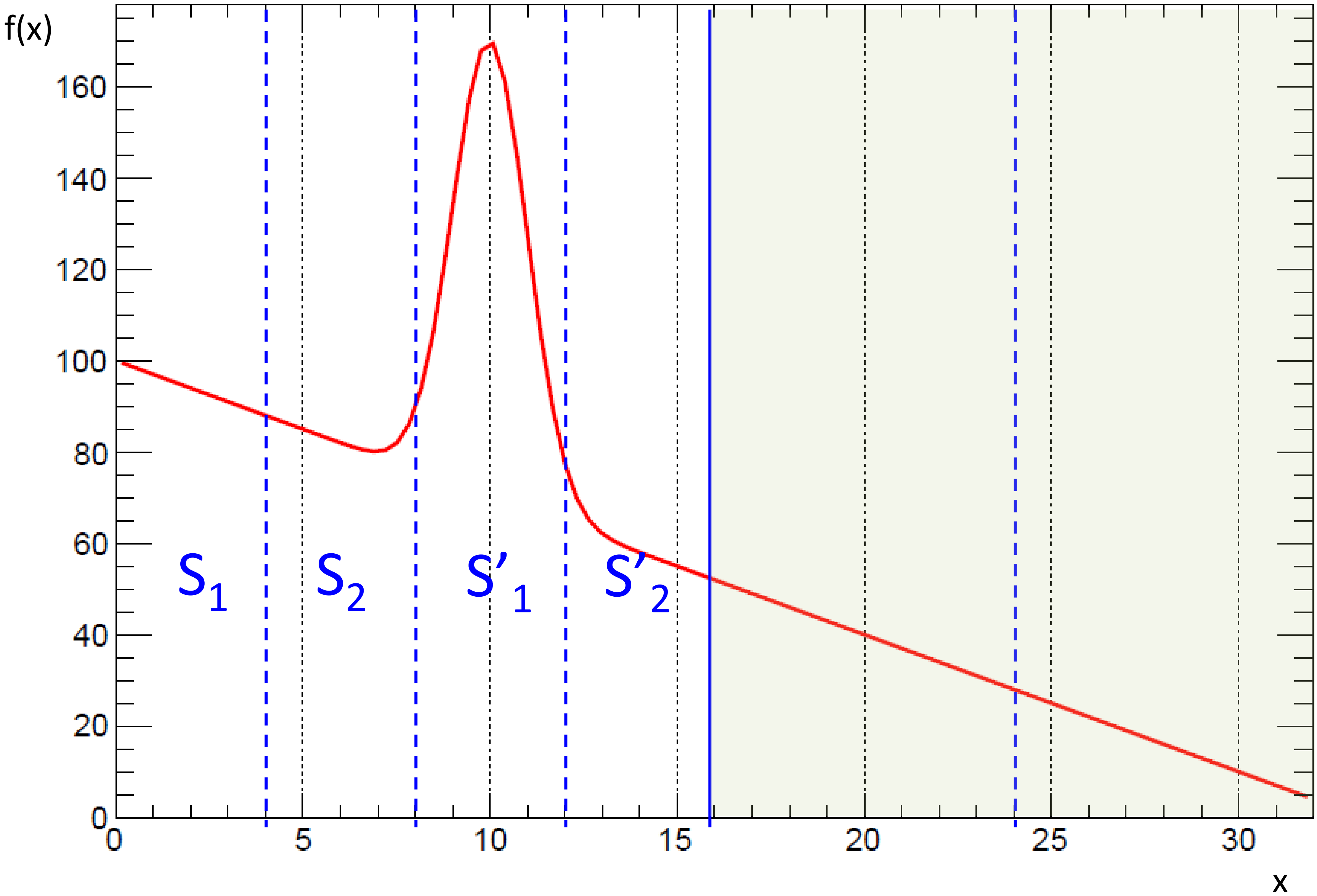
Αριθμητική Ολοκλήρωση: Προσαρμοζόμενες μέθοδοι

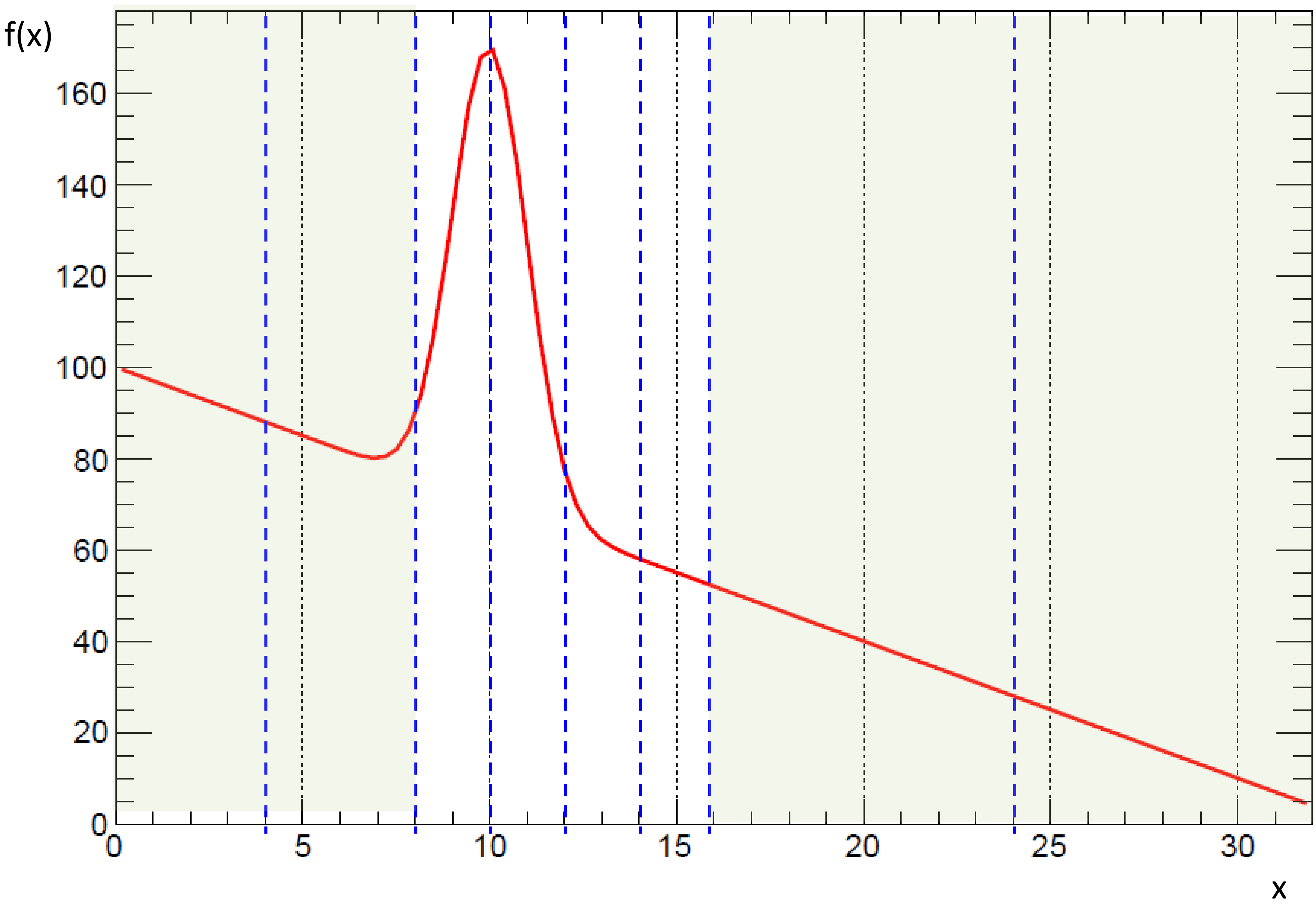


$$\frac{S_1 + S_2 - S_0}{15} < (?)eps/2$$

$$\frac{S'_1 + S'_2 - S'_0}{15} < (?)eps/2$$

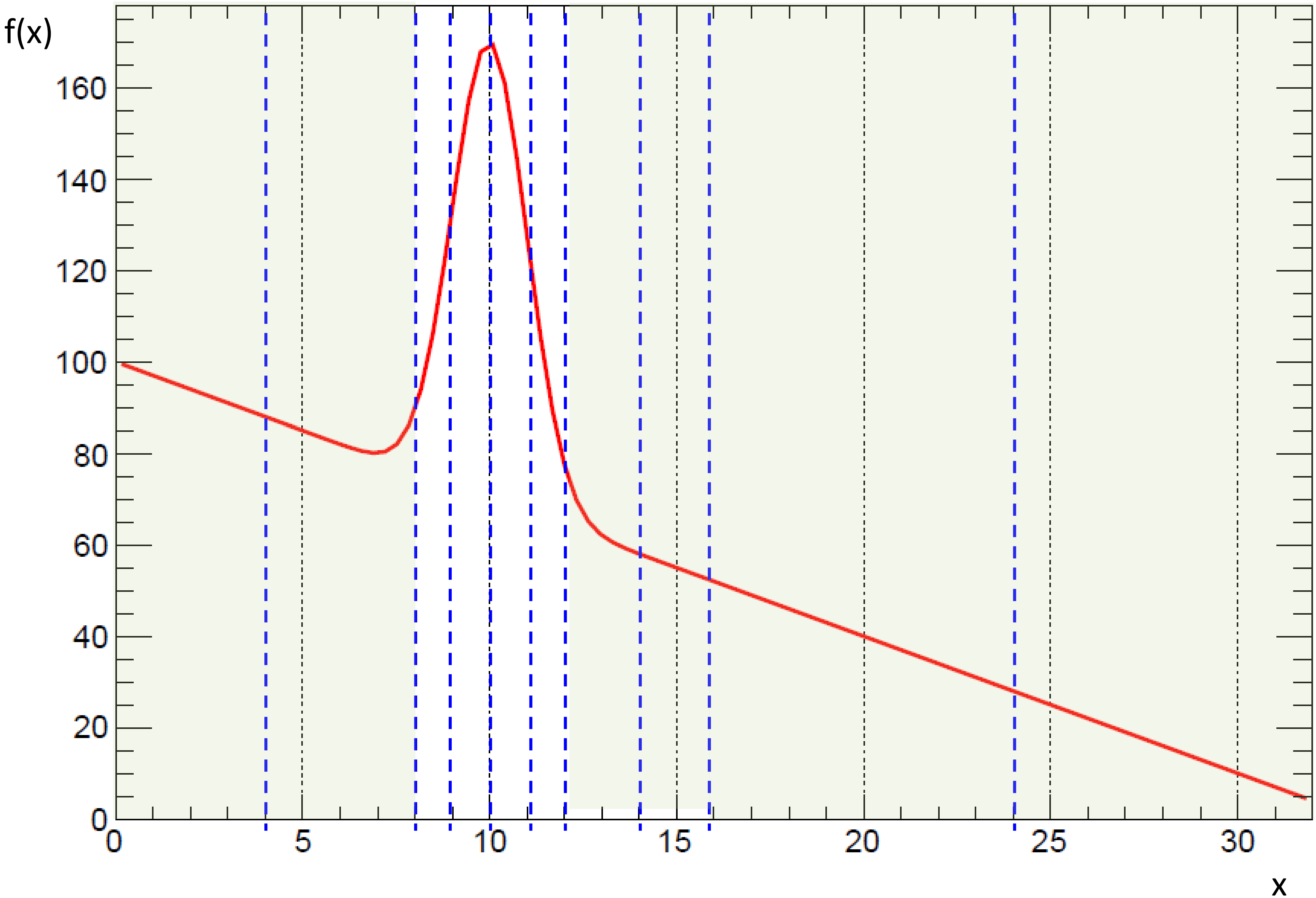
Αριθμητική Ολοκλήρωση: Προσαρμοζόμενες μέθοδοι



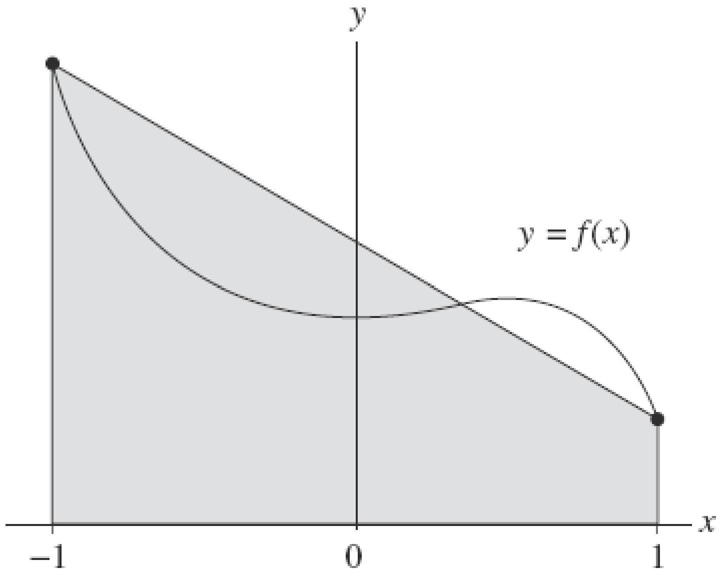


f(x)
160
140
120
100
80
60
40
20
0
0
5
10
15
20
25
30
x

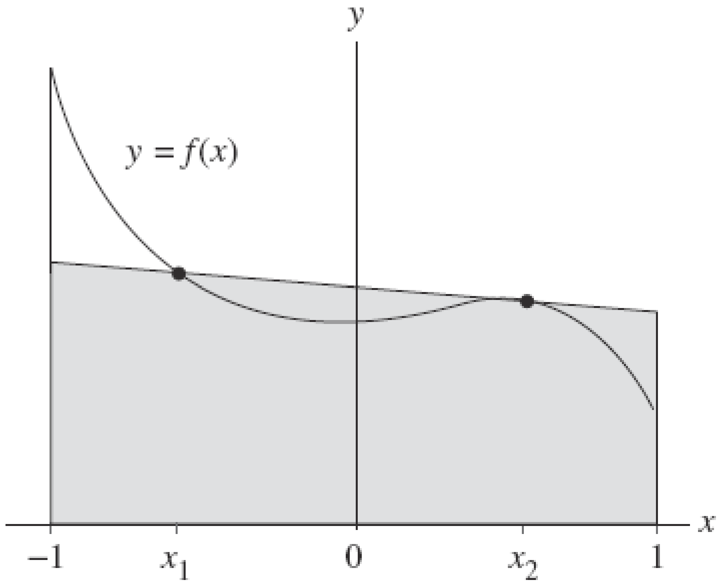
Αριθμητική Ολοκλήρωση: Προσαρμοζόμενες μέθοδοι



Κίνητρο



Κανόνας τραπεζίου με χρήση των σημείων -1 και +1



Κανόνας τραπεζίου με χρήση των σημείων x_1 και x_2

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n} c_i f(x_i)$$

2n ελεύθερες παράμετροι → Μπορεί να επιτευχθεί βαθμός ακρίβειας 2n-1

Χρησιμοποιώ βάση ορθογωνίων πολυωνύμων

$$\int_a^b \Phi_i(x)\Phi_j W(x)dx = \begin{cases} 0, & i \neq j \\ \varepsilon_i > 0, & i = j \end{cases}$$

Συνάρτηση βάρους

όχι πάντα απαραίτητη

π.χ. Πολυώνυμα Legendre, ορθογώνια στο [-1,+1]

$$P_n(x) = \frac{1}{2^n n!}\frac{d^n}{dx^n}(x^2-1)^n$$

Πολυώνυμα Legendre, ορθογώνια στο [-1,+1] με συνάρτηση βάρους W(x)=1

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2 - 1)^n$$

$$P_0(x) = 1$$

$$P_1(x) = x$$

$$P_2(x) = x^2 - \frac{1}{3}$$

$$P_3(x) = x^3 - \frac{3}{5}x$$

$$P_4(x) = x^4 - \frac{6}{7}x^2 + \frac{3}{35}$$

Έστω πολυώνυμο βαθμού k < 2n

$$P(x)=Q(x)P_n(x)+R(x)$$

$$\int_{-1}^{+1} P(x)dx = \cancel{\int_{-1}^{+1} Q(x)P_n(x)dx} + \int_{-1}^{+1} R(x)dx \approx \sum_{i=1}^{n} c_i R(x_i)$$

Λόγω ορθογωνιότητας
Των πολυωνύμων βάσης

Έστω πολυώνυμο βαθμού $k < 2n$ $\quad$ $P(x)=Q(x)P_n(x)+R(x)$

$$\int_{-1}^{+1} P(x)dx = \cancel{\int_{-1}^{+1} Q(x)P_n(x)dx} + \int_{-1}^{+1} R(x)dx \quad \approx \sum_{i=1}^{n} c_i R(x_i)$$

Αλλά $\quad P(x_i)=Q(x_i)P_n(x_i)+R(x_i)$

Οπότε θέτοντας ως x_i τις ρίζες του $P_n(x)$

Έστω πολυώνυμο βαθμού k < 2n $P(x)=Q(x)P_n(x)+R(x)$

$$\int_{-1}^{+1} P(x)dx = \cancel{\int_{-1}^{+1} Q(x)P_n(x)dx} + \int_{-1}^{+1} R(x)dx \approx \sum_{i=1}^{n} a_i R(x_i)$$

Αλλά $P(x_i)=Q(x_i)P_n(x_i)+R(x_i)$

Οπότε θέτοντας ως x_i τις ρίζες του $P_n(x)$

$Q(x_i)P_n(x_i) = 0 \rightarrow R(x_i)=P(x_i)$

$$\Rightarrow \boxed{\int_{-1}^{1} P(x)dx \approx \sum_{i=1}^{n} c_i P(x_i)}$$

Όπου $\boldsymbol{x_i}$ οι ρίζες του πολυωνύμου Legendre βαθμού $\boldsymbol{n}$

Επομένως προσεγγίζουμε το ολοκλήρωμα ως:

$$\boxed{\int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{n} c_i f(x_i)}$$

όπου x_i οι ρίζες του πολυωνύμου Legendre Βαθμού n

και για τον υπολογισμό των c_i επιλύουμε το σύστημα:

$$\begin{aligned}
b_0 &= \int_{-1}^{+1} P_0^2(x)dx &&= \sum_{i=1}^{n} c_i P_0^2(x_i) \\
b_1 &= \int_{-1}^{+1} P_1^2(x)dx &&= \sum_{i=1}^{n} c_i P_1^2(x_i) \\
\ldots & \quad \ldots && \quad \ldots \\
b_{n-1} &= \int_{-1}^{+1} P_{n-1}^2(x)dx &&= \sum_{i=1}^{n} c_i P_{n-1}^2(x_i)
\end{aligned}$$

Αλλά, δε χρειάζεται να λύσουμε εμείς τα συστήματα, καθώς υπάρχουν πίνακες με τις ρίζες των πολυωνύμων Legendre και τους συντελεστές $c_{n,i}$ ($w_{N,k}$ στον παρακάτω πίνακα)

$$\int_{-1}^{1} f(x)\,dx = \sum_{k=1}^{N} w_{N,k} f(x_{N,k}) + E_N(f)$$

N	Abscissas, $x_{N,k}$	Weights, $w_{N,k}$	Truncation error, $E_N(f)$
2	−0.5773502692 0.5773502692	1.0000000000 1.0000000000	$\frac{f^{(4)}(c)}{135}$
3	±0.7745966692 0.0000000000	0.5555555556 0.8888888888	$\frac{f^{(6)}(c)}{15{,}750}$
4	±0.8611363116 ±0.3399810436	0.3478548451 0.6521451549	$\frac{f^{(8)}(c)}{3{,}472{,}875}$
5	±0.9061798459 ±0.5384693101 0.0000000000	0.2369268851 0.4786286705 0.5688888888	$\frac{f^{(10)}(c)}{1{,}237{,}732{,}650}$
6	±0.9324695142 ±0.6612093865 ±0.2386191861	0.1713244924 0.3607615730 0.4679139346	$\frac{f^{(12)}(c)2^{13}(6!)^4}{(12!)^3 13!}$
7	±0.9491079123 ±0.7415311856 ±0.4058451514 0.0000000000	0.1294849662 0.2797053915 0.3818300505 0.4179591837	$\frac{f^{(14)}(c)2^{15}(7!)^4}{(14!)^3 15!}$
8	±0.9602898565 ±0.7966664774 ±0.5255324099 ±0.1834346425	0.1012285363 0.2223810345 0.3137066459 0.3626837834	$\frac{f^{(16)}(c)2^{17}(8!)^4}{(16!)^3 17!}$

Ένα ολοκλήρωμα στο διάστημα [α,b] μπορεί να μετατραπεί σε ολοκλήρωμα στο διάστημα [-1,1] κάνοντας απλά την αλλαγή μεταβλητών:

$$t = \frac{2x - a - b}{b - a} \Leftrightarrow x = \frac{1}{2}\left[(b-a)t + a + b\right]$$

Οπότε:

$$\int_a^b f(x)dx = \int_{-1}^{1} f\left(\frac{(b-a)t + a + b}{2}\right)\left(\frac{b-a}{2}\right)dt$$

Αριθμητική Ολοκλήρωση: Μέθοδοι Gauss – Παράδειγμα Ι

Να υπολογιστεί το ολοκλήρωμα: $$\int_1^3 x^6 - x^2 \sin(2x)dx = 317.344$$

με τις μεθόδους που πραγματοποιούν δύο δειγματοληψίες της *f(x)*

Newton-cotes κλειστή (τραπεζίου)

$$I \approx \frac{2}{2}\left[f(1) + f(3)\right] = \underline{731.605}$$

Newton-cotes ανοιχτή 2 σημείων

$$I \approx \frac{3(2/3)}{2}\left[f(\frac{5}{3}) + f(\frac{7}{3})\right] = \underline{188.786}$$

Gauss 2 σημείων

$$\int_a^b f(x)\,dx = \int_{-1}^1 f\left(\frac{(b-a)t+(b+a)}{2}\right)\frac{(b-a)}{2}\,dt$$

$$\int_1^3 x^6 - x^2 \sin(2x)dx = \int_{-1}^1 (t+2)^6 - (t+2)^2 \sin\big(2(t+2)\big)dt$$

$$I \approx \left[f(-\frac{1}{\sqrt{3}}+2) + f(\frac{1}{\sqrt{3}}+2)\right] = \underline{306.820}$$

Αριθμητική Ολοκλήρωση: Μέθοδοι Gauss – Παράδειγμα II

Να υπολογιστεί το ολοκλήρωμα: $\int_1^3 x^6 - x^2 \sin(2x)dx = 317.344$

με τις μεθόδους που πραγματοποιούν τρεις δειγματοληψίες της *f(x)*

Newton-cotes κλειστή (Simpson)

$$I \approx \frac{1}{3}\left[f(1) + 4f(2) + f(3)\right] = \underline{333.238}$$

Newton-cotes ανοιχτή 3 σημείων

$$I \approx \frac{4(1/2)}{3}\left[2f(1.5) + f(2) + 2f(2.5)\right] = \underline{303.591}$$

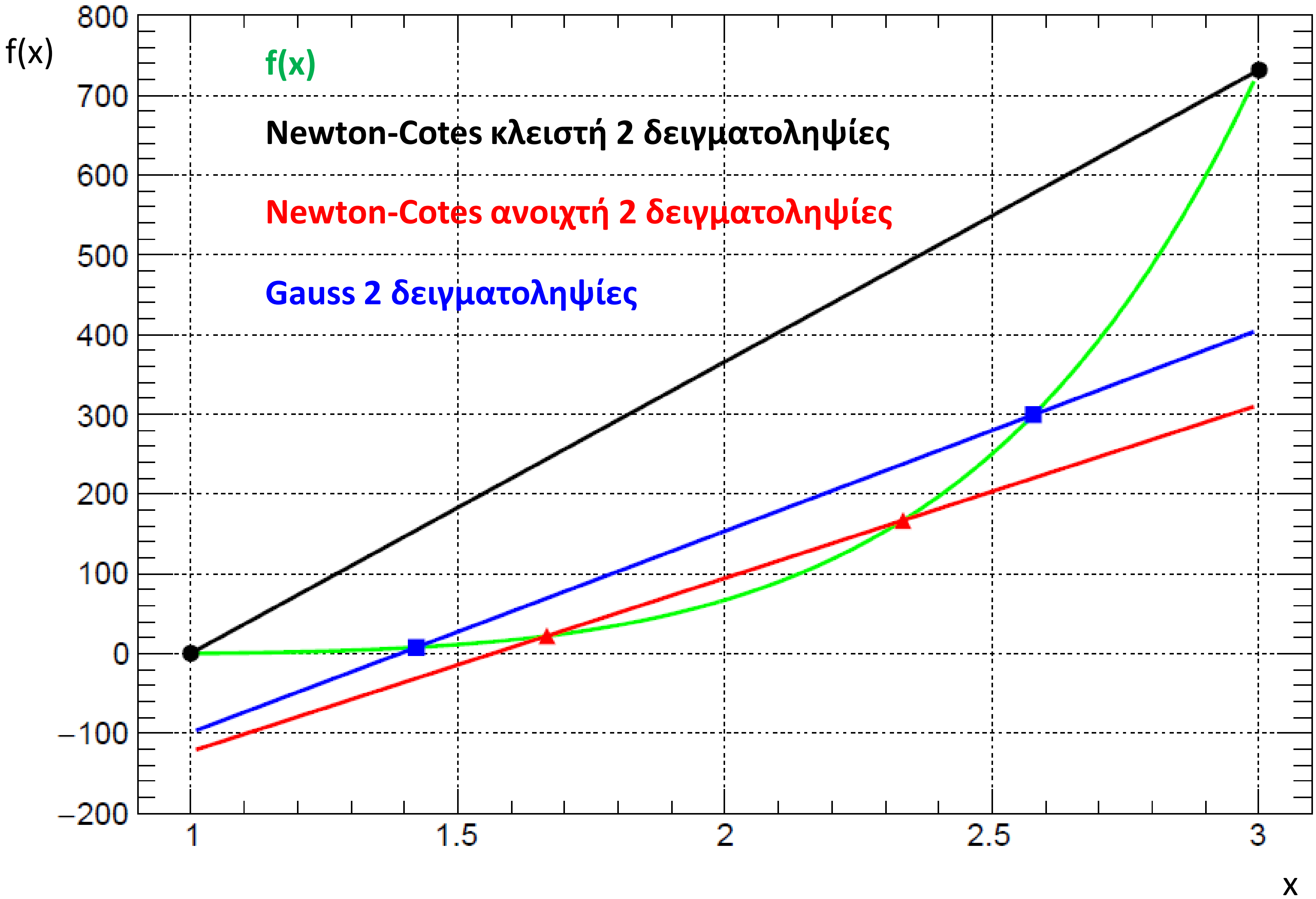
Gauss 2 σημείων

$$\int_a^b f(x)\,dx = \int_{-1}^1 f\left(\frac{(b-a)t + (b+a)}{2}\right)\frac{(b-a)}{2}\,dt$$

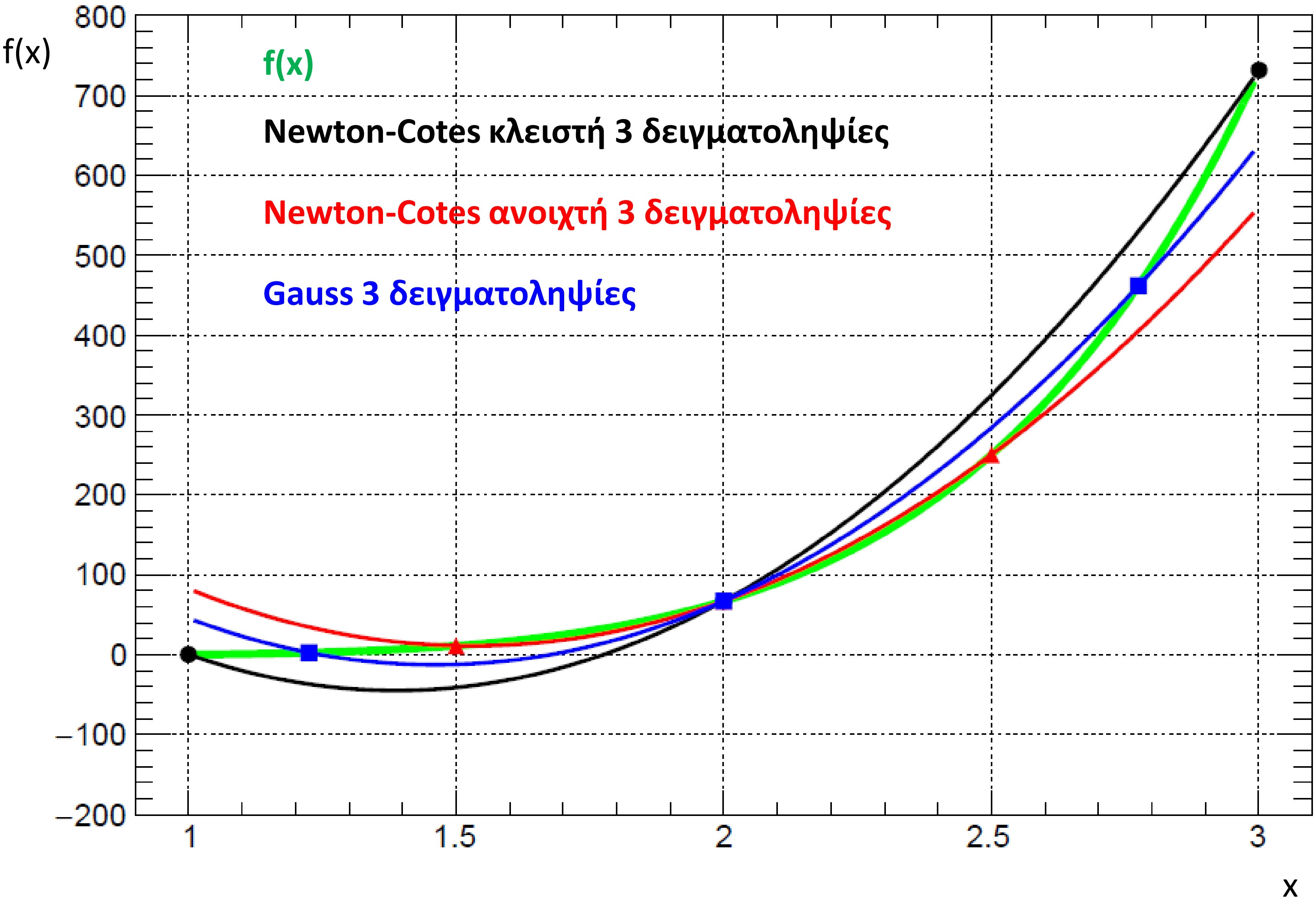
$$\int_1^3 x^6 - x^2 \sin(2x)dx = \int_{-1}^1 (t+2)^6 - (t+2)^2 \sin\big(2(t+2)\big)dt$$

$$I \approx \left[0.\overline{5}\, f(-\sqrt{\frac{3}{5}} + 2) + 0.\overline{8}\, f(2) + 0.\overline{5}\, f(\sqrt{\frac{3}{5}} + 2)\right] = \underline{317.264}$$

Αριθμητική Ολοκλήρωση: Μέθοδοι Gauss – Παράδειγμα III



Αριθμητική Ολοκλήρωση: Μέθοδοι Gauss – Παράδειγμα IV



Αριθμητική Ολοκλήρωση: Μέθοδος Gauss 3 σημείων

$$P_3(x) = x^3 - \frac{3}{5}x$$

$$\int_a^b f(x)dx = \int_{-1}^{1} f\left(\frac{(b-a)t+a+b}{2}\right)\left(\frac{b-a}{2}\right)dt$$

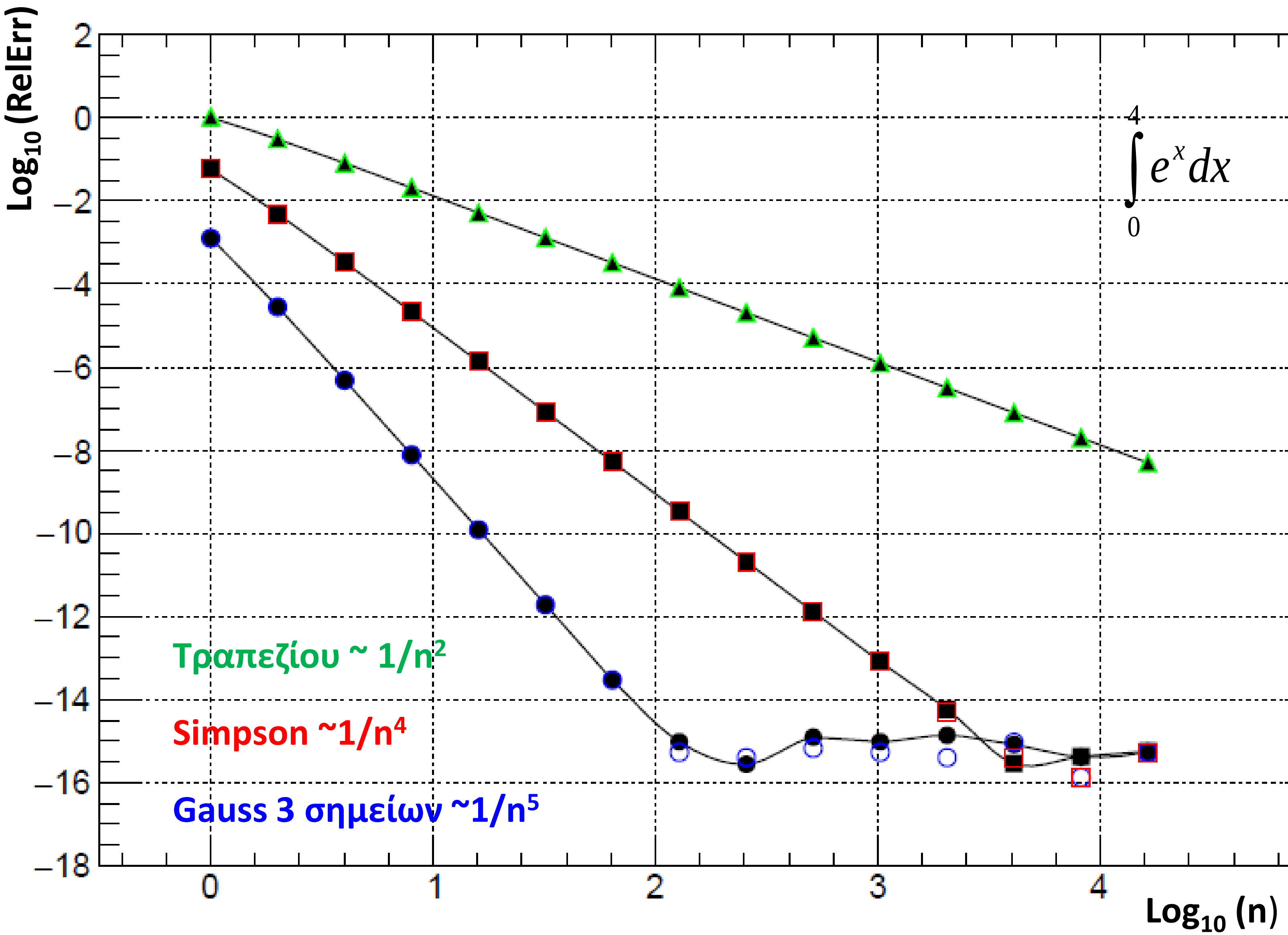
```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

double myfunc(double x)
{
   double ff=0.;
   ff=exp(x);
   //ff= pow(x,6)-x*x*sin(2*x);
   return ff;
}

double gausquad3(double a, double b)
{
   double integ=0.;
   double t[3]={-pow(3./5.,0.5), 0, pow(3./5.,0.5)};
   double x[3]={0.5*((b-a)*t[0]+b+a), 0.5*((b-a)*t[1]+b+a), 0.5*((b-a)*t[2]+b+a)};
   double integ=0.5*(b-a)*(myfunc(x[0])/1.8+myfunc(x[1])/1.125+myfunc(x[2])/1.8);
   return integ;
}
```

Αριθμητική Ολοκλήρωση: Μέθοδος Gauss 3 σημείων

```
void gausquad(double a, double b, int n)
{
  double h=(b-a)/n;
  double integ=0;
  for (int i=0; i<n; i++)
     {
       double x0=a+i*h;
       double x1=x0+h;
       integ+=gausquad3(x0,x1);
     }
  double h=(b-a)/2/n;
  double integ2=0;
  for (int i=0; i<2*n; i++)
     {
       double x0=a+i*h;
       double x1=x0+h;
       integ2+=gausquad3(x0,x1);
     }
  double einteg=32.*fabs(integ-integ2)/31.;
  double trueval=exp(4.)-1.;
  printf("integral %3.12f +- %3.12f deviation %3.12f", integ, einteg, trueval-integ;
}
```

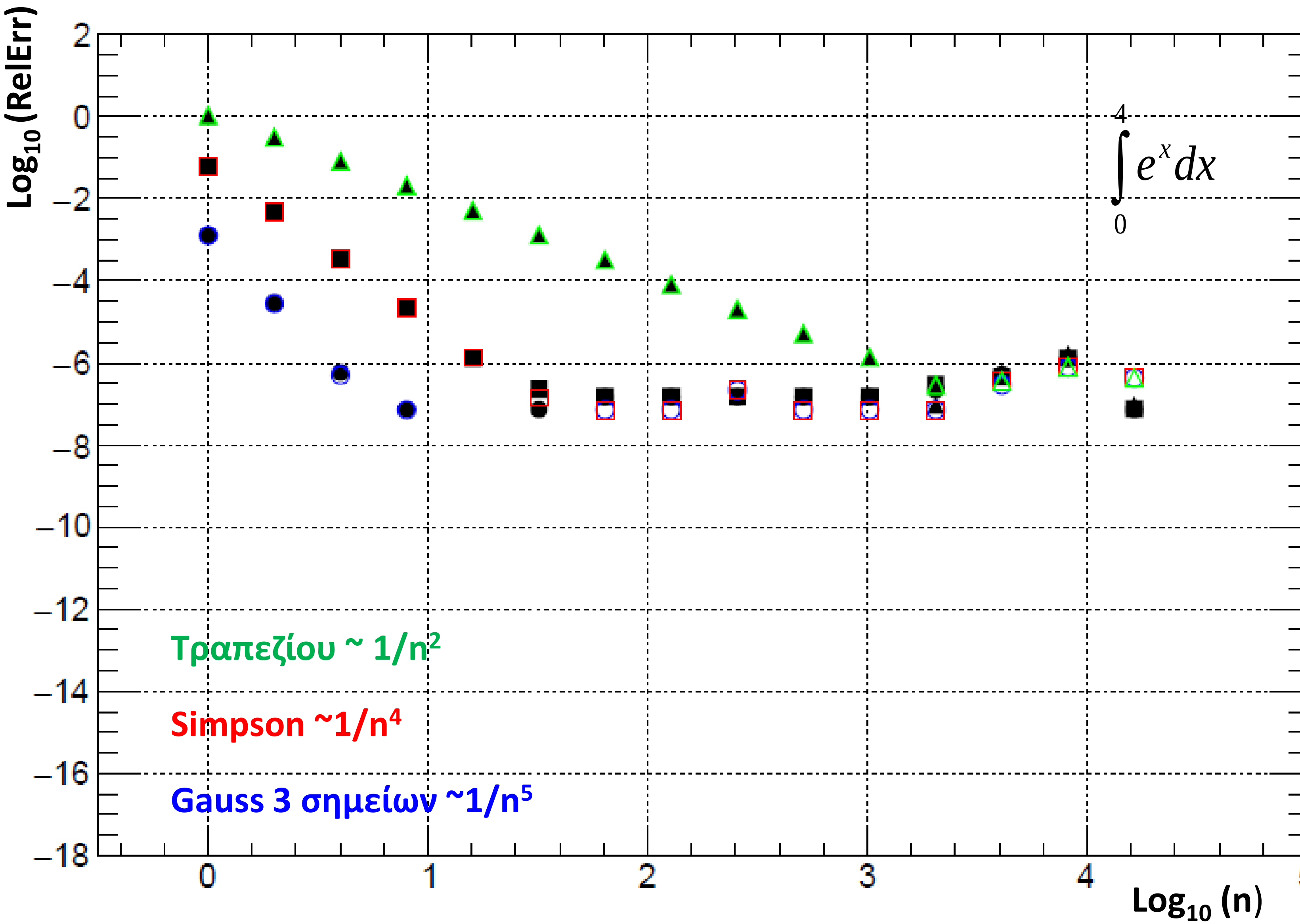
Αριθμητική Ολοκλήρωση: Σύγκριση Μεθόδων

Χρήση **double** (1 bit for the sign, 11 bits for the exponent, and 52* bits for the value)



Αριθμητική Ολοκλήρωση: Σύγκριση Μεθόδων

Χρήση **float** (1 bit for the sign, 8 bits for the exponent, and 23* for the value)



- Διάστημα [α,b] $\Rightarrow$ Legendre

$$P_n(x) = \frac{1}{2^n n!}\frac{d^n}{dx^n}(x^2-1)^n$$

- Διάστημα [0,∞] $\Rightarrow$ Laguerre

$$L_n(x) = \frac{e^x}{n!}\frac{d^n}{dx^n}(x^n e^{-x})$$

- Διάστημα [-∞, +∞] $\Rightarrow$ Hermite

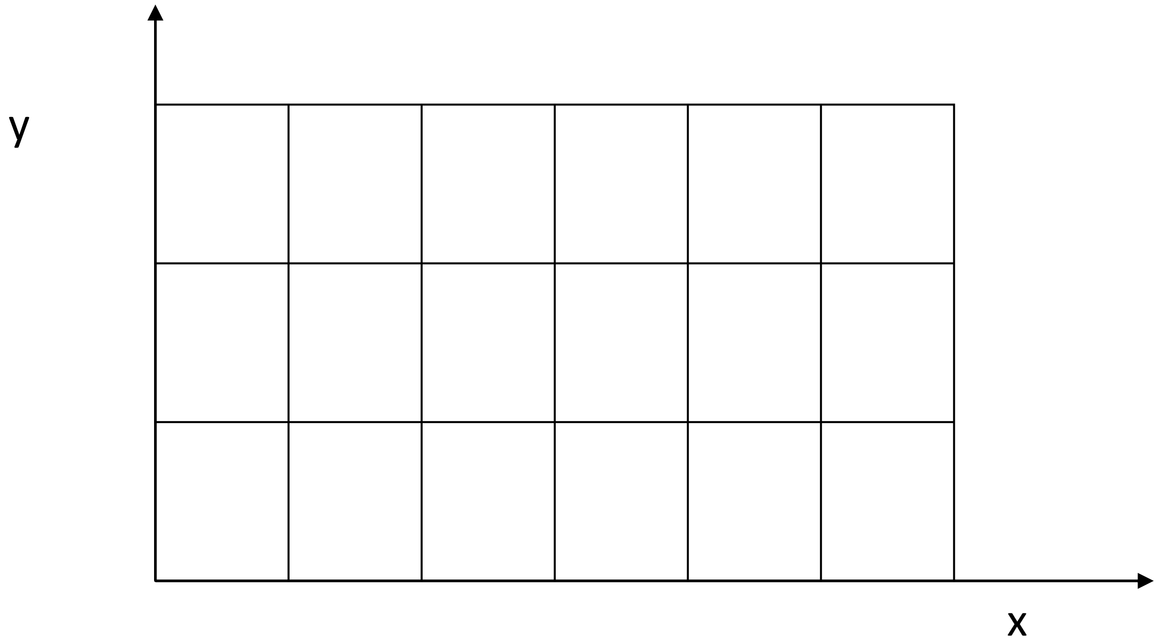
$$H_n(x) = (-1)^n e^{x^2}\frac{d^n}{dx^n}e^{-x^2}$$

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c  gauss.f: Points and weights for Gaussian quadrature
c  taken from: "Projects in Computational Physics" by Landau and Paez
c                          copyrighted by John Wiley and Sons, New York
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     rescale rescales the gauss-legendre grid points and weights
c     npts     number of points
c     job = 0  rescalling uniformly between (a,b)
c           1  for integral (0,b) with 50% points inside (0, ab/(a+b))
c           2  for integral (a,inf) with 50% inside (a,b+2a)
c     x, w     output grid points and weights.
c
      subroutine gauss(npts,job,a,b,x,w)
      integer npts,job,m,i,j
      real*8 x(npts),w(npts),a,b,xi
      real*8 t,t1,pp,p1,p2,p3,aj
      real*8 eps,pi,zero,two,one,half,quarter
      parameter (pi = 3.14159265358979323846264338328, eps = 3.0E-14)
      parameter (zero=0.0d0,one=1.0d0,two=2.0d0)
      parameter (half=0.5d0,quarter=0.25d0)

      m=(npts+1)/2
      do 1020 i=1,m
        t=cos(pi*(i-quarter)/(npts+half))
 1000   continue
        p1=one
        p2=zero
        aj=zero
        do 1010 j=1,npts
          p3=p2
          p2=p1
          aj=aj+one
          p1=((two*aj-one)*t*p2-(aj-one)*p3)/aj
 1010   continue
        pp=npts*(t*p1-p2)/(t*t-one)
        t1=t
        t=t1-p1/pp
c
        if(abs(t-t1).gt.eps) goto 1000
c
        x(i)=-t
        x(npts+1-i)=t
        w(i)=two/((one-t*t)*pp*pp)
        w(npts+1-i)=w(i)
 1020 continue
c
c rescale the grid points
      if (job.eq.0) then
c     scale to (a,b) uniformly
        do 1030 i=1,npts
          x(i)=x(i)*(b-a)/two+(b+a)/two
          w(i)=w(i)*(b-a)/two
 1030   continue
      elseif (job.eq.1) then
c scale to (0,b) with 50% points inside (0,ab/(a+b))
        do 1040 i=1,npts
          xi=x(i)
          x(i)=a*b*(one+xi)/(b+a-(b-a)*xi)
          w(i)=w(i)*two*a*b*b/((b+a-(b-a)*xi)*(b+a-(b-a)*xi))
 1040   continue
      elseif (job.eq.2) then
c scale to (a,inf) with 50% points inside (a,b+2a)
        do 1050 i=1,npts
          xi=x(i)
          x(i)=(b*xi+b+a+a)/(one-xi)
          w(i)=w(i)*two*(a+b)/((one-xi)*(one-xi))
 1050   continue
      else
        pause 'Wrong value of job'
      endif
c
      return
      end
```

$$\iint_R f(x,y)dA = \int_a^b dx\left(\int_c^d f(x,y)dy\right)$$

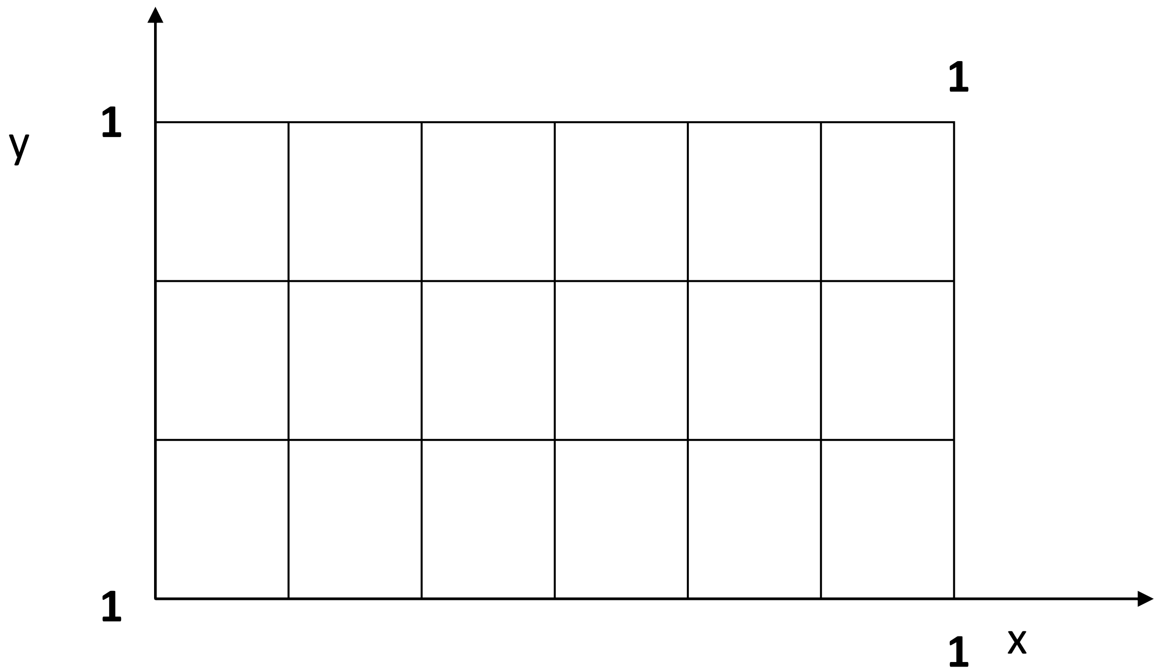
Παρατήρηση:

Δεν απαιτείται ίση διαμέριση σε κάθε διάσταση



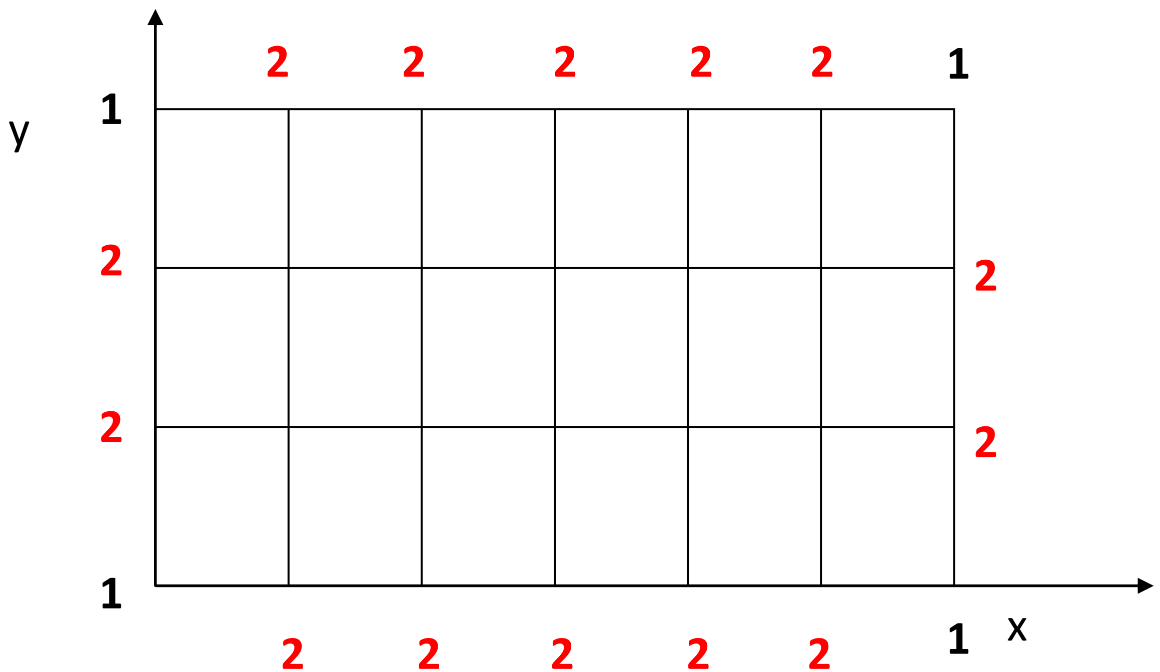
$$\iint_R f(x,y)dA = \int_a^b dx \int_c^d f(x,y)dy$$

Μέθοδος Τραπεζίου



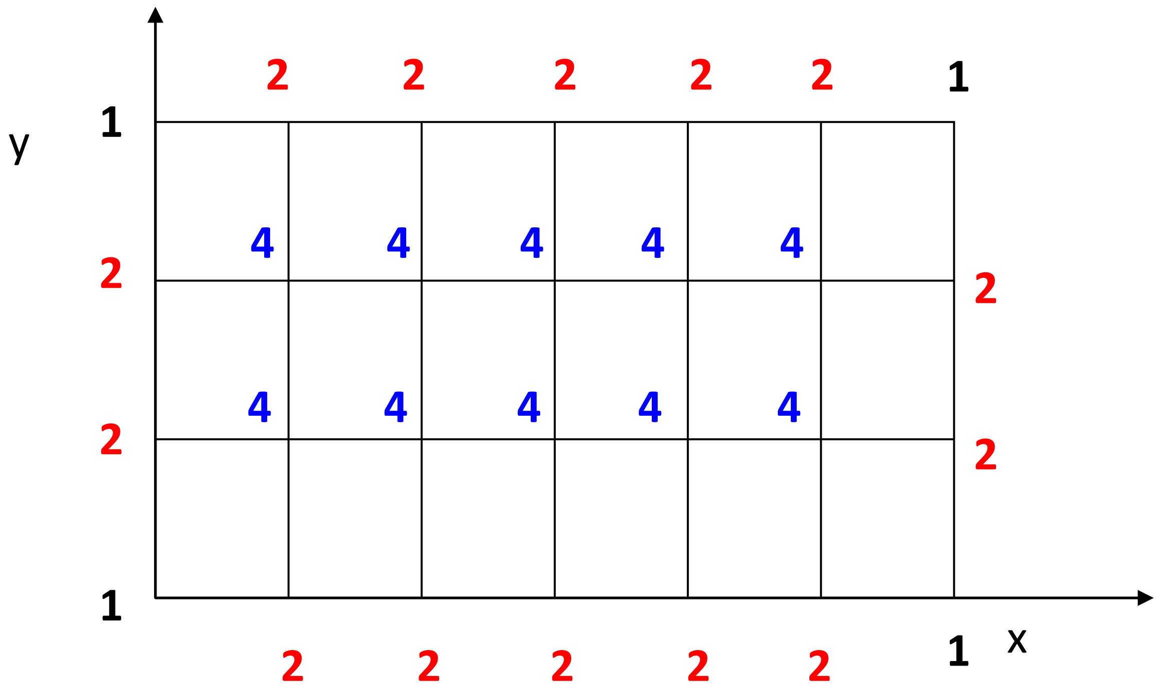
$$\iint_R f(x,y)dA = \int_a^b dx \int_c^d f(x,y)dy$$

Μέθοδος Τραπεζίου



$$\iint_R f(x,y)\,dA = \int_a^b dx \int_c^d f(x,y)\,dy$$

Μέθοδος Τραπεζίου



$$\iint_R f(x,y)dA = \int_a^b dx \int_c^d f(x,y)dy$$

Μέθοδος Τραπεζίου σε δύο διαστάσεις

Για n διαστήματα στη διεύθυνση x και m διαστήματα στην διεύθυνση y, θέτοντας ***h=(b-a)/n, k=(d-c)/m***

$$\begin{aligned}\int_a^b dx \int_c^d f(x,y)dy \quad &= \frac{hk}{4}\{ f(x_0,y_0)+f(x_0,y_m)+f(x_n,y_0)+f(x_n,y_m) \\ &+2\sum_{j=1}^{m-1} f(x_0,y_j)+2\sum_{j=1}^{m-1} f(x_n,y_j)+2\sum_{i=1}^{n-1} f(x_i,y_0)+2\sum_{i=1}^{n-1} f(x_i,y_m) \\ &+4\sum_{i=1}^{n-1}\sum_{j=1}^{m-1} f(x_i,y_j)\}\end{aligned}$$

Μέθοδος Simpson σε δύο διαστάσεις

Θέτοντας ***h=(b-a)/n, k=(d-c)/m***

$$\int_a^b \int_c^d f(x,y)\,dy\,dx \approx \frac{hk}{9}\Bigg\{\Bigg[f(x_0,y_0) + 2\sum_{i=1}^{(n/2)-1} f(x_{2i},y_0)$$

$$+ 4\sum_{i=1}^{n/2} f(x_{2i-1},y_0) + f(x_n,y_0)\Bigg]$$

$$+ 2\Bigg[\sum_{j=1}^{(m/2)-1} f(x_0,y_{2j}) + 2\sum_{j=1}^{(m/2)-1}\sum_{i=1}^{(n/2)-1} f(x_{2i},y_{2j})$$

$$+ 4\sum_{j=1}^{(m/2)-1}\sum_{i=1}^{n/2} f(x_{2i-1},y_{2j}) + \sum_{j=1}^{(m/2)-1} f(x_n,y_{2j})\Bigg]$$

$$+ 4\Bigg[\sum_{j=1}^{m/2} f(x_0,y_{2j-1}) + 2\sum_{j=1}^{m/2}\sum_{i=1}^{(n/2)-1} f(x_{2i},y_{2j-1})$$

$$+ 4\sum_{j=1}^{m/2}\sum_{i=1}^{n/2} f(x_{2i-1},y_{2j-1}) + \sum_{j=1}^{m/2} f(x_n,y_{2j-1})\Bigg]$$

$$+ \Bigg[f(x_0,y_m) + 2\sum_{i=1}^{(n/2)-1} f(x_{2i},y_m) + 4\sum_{i=1}^{n/2} f(x_{2i-1},y_m) + f(x_n,y_m)\Bigg]\Bigg\}.$$

Μέθοδος Simpson σε δύο διαστάσεις

Θέτοντας ***h=(b-a)/n, k=(d-c)/m***

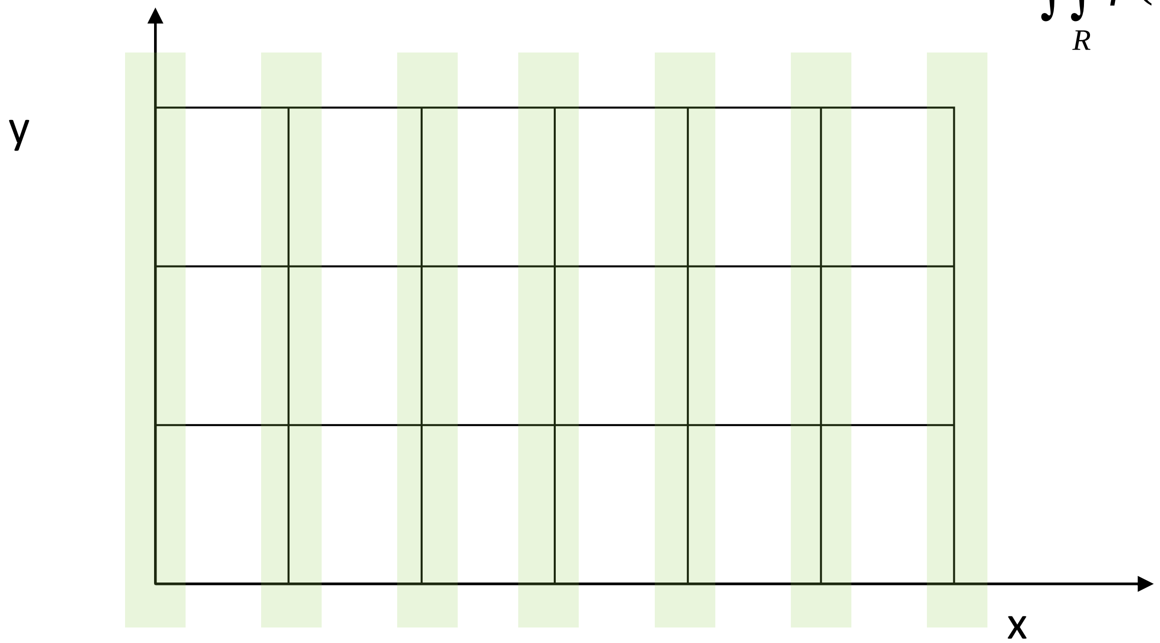
Η αβεβαιότητα είναι:

$$E = -\frac{(d-c)(b-a)}{180}\left[h^4 \frac{\partial^4 f}{\partial x^4}(\overline{\eta}, \overline{\mu}) + k^4 \frac{\partial^4 f}{\partial y^4}(\hat{\eta}, \hat{\mu})\right]$$

Για ευκολία στον προγραμματισμό μπορούμε να υπολογίζουμε το ολοκλήρωμα σε μία διάσταση χρησιμοποιώντας ως συνάρτηση προς ολοκλήρωση την αριθμητική ολοκλήρωση στην άλλη διάσταση

Με αυτόν τον τρόπο κάνουμε περισσότερες δειγματοληψίες της f, αλλά η γενίκευση είναι πολύ απλή και μπορεί να υλοποιηθεί για οποιαδήποτε μέθοδο σε οποιεσδήποτε διστάσεις

$$\int\int_R f(x,y)dA = \int_a^b dx \int_c^d f(x,y)dy$$

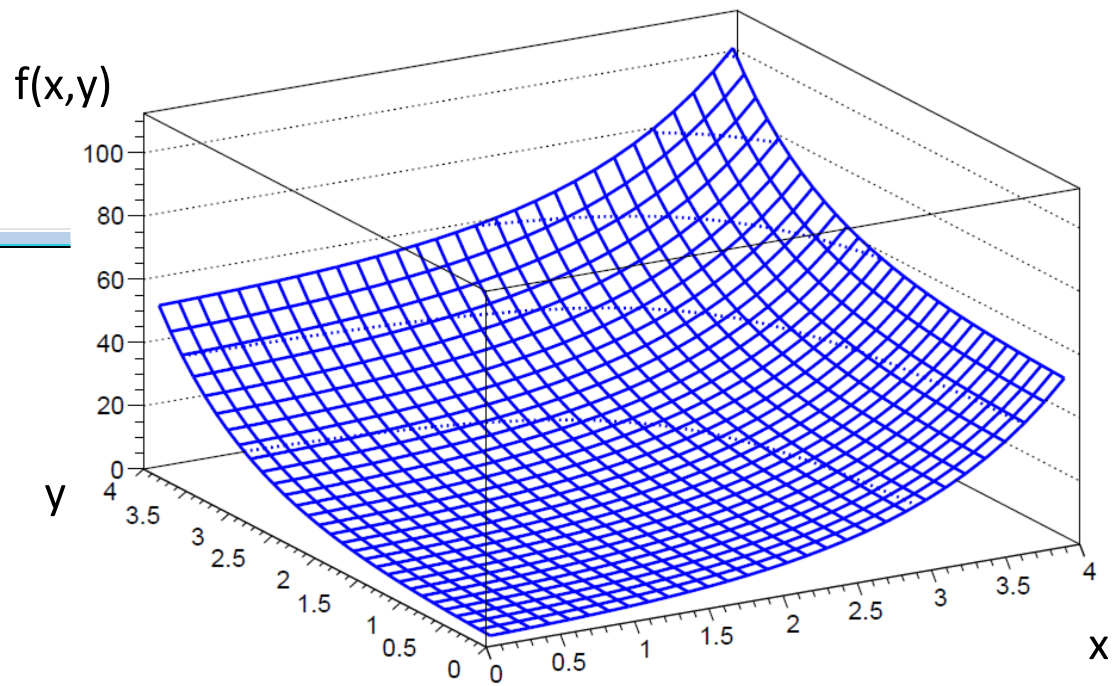


```
double myfunc(double x, double y){
  double ff=0.;
  ff=exp(x)+exp(y);
  return ff;
}
double simpsony(double xx, double a, double b, int n)
{// Υπολογισμός ολοκληρώματος στη διάσταση y
double h=(b-a)/2/n;
double integral_b=0, integral_c=0;
double integral_a=(myfunc(xx,a)+myfunc(xx,b));
for (int i=1; i<2*n; i++){
    double yy=a+h*i;
    if(i%2==0){integral_b+=myfunc(xx,yy);}
    else {integral_c+=myfunc(xx,yy);}
    }
double integ=(h/3.)*(integral_a+2*integral_b+4*integral_c);
return integ;
}
void integral2d(double a, double b, double c, double d, int n, int m)
{ // Υπολογισμός ολοκληρώματος στη διάσταση x
    double h=(b-a)/2/n;
    double integral_b=0, integral_c=0;
    double integral_a=(simpsony(a, c, d, m)+simpsony(b, c, d, m) );
    for (int i=1; i<2*n; i++){
         double xx=a+h*i;
         if(i%2==0){integral_b+=simpsony(xx, c, d, m);}
         else {integral_c+=simpsony(xx, c, d, m);}
         }
    double integ=(h/3.)*(integral_a+2*integral_b+4*integral_c);
    double trueval=8.*(exp(4.)-1.);
    cout << integ<<" "<<trueval<<endl;
}
```

Αριθμητική Ολοκλήρωση: Πολλαπλά Ολοκληρώματα

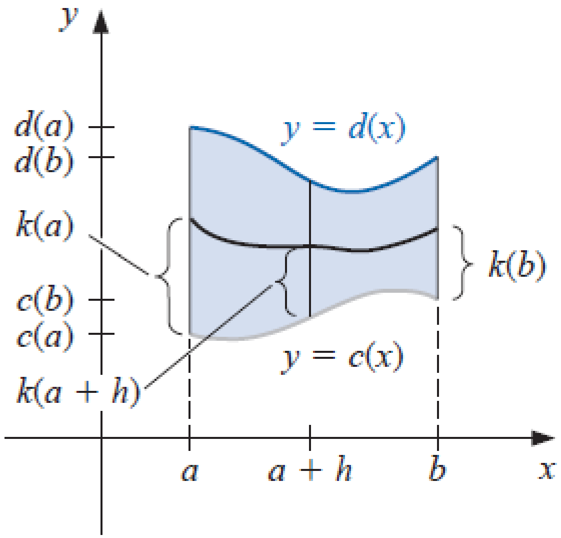
ROOT session

```
root [41]
root [41]
root [41]
root [41]
root [41]
root [41] .x integral2d.c(0.,4.,0.,4.,1,1)
454.157 428.785
root [42] .x integral2d.c(0.,4.,0.,4.,2,2)
430.911 428.785
root [43] .x integral2d.c(0.,4.,0.,4.,4,4)
428.93 428.785
root [44] .x integral2d.c(0.,4.,0.,4.,8,8)
428.794 428.785
root [45] .x integral2d.c(0.,4.,0.,4.,16,16)
428.786 428.785
root [46]
root [46]
root [46]
root [46]
root [46]
root [46]
root [46]
root [46]
root [46]
root [46]
```



Χρησιμοποιώντας μια παραλλαγή του προηγούμενου υπολογιστικού κώδικα μπορούμε να υπολογίσουμε ολοκληρώματα χωρίς την απαίτηση ορθογώνιων συνοριακών συνθηκών

$$\int_a^b \int_{c(x)}^{d(x)} f(x,y)\,dy\,dx \quad \text{ή} \quad \int_c^d \int_{a(y)}^{b(y)} f(x,y)\,dx\,dy$$



Αρκεί να αλλάξουμε τις κλήσεις
`simpsony(xx, c, d, m)`
με
`simpsony(xx, c(xx), d(xx), m)`
αφού ορίσουμε πρώτα κατάλληλα τις συναρτήσεις αυτές σύμφωνα με τις απαιτήσεις του προβλήματος

Παρατήρηση:
Η ακρίβεια (ή ταχύτητα σύγκλισης) ως συνάρτηση των δειγματοληψιών της *f(x)* μειώνεται αυξάνοντας τις διαστάσεις του ολοκληρώματος

Μέθοδος	1 διάσταση	d διαστάσεις
Τραπεζίου	N^{-2}	$N^{-2/d}$
Simpson	N^{-4}	$N^{-4/d}$
Gauss τάξης m	N^{-2m+1}	$N^{(-2m+1)/d}$
Monte Carlo	$N^{-1/2}$	$N^{-1/2}$

Άσκηση Α 1

Κατασκευάστε υπολογιστικούς κώδικες που να υπολογίζουν ολοκληρώματα σε 3 διαστάσεις σε (i) καρτεσιανές και (ii) σε πολικές συντεταγμένες. Χρησιμοποιείστε μια μέθοδο Gauss.

α) Υπολογίστε τη μάζα και το κέντρο βάρους κύβου μοναδιαίας ακμής και πυκνότητας $\rho(x,y,z)= 1+x^2+2y^2+3z^2$ με ακρίβεια 1‰. Θεωρήστε ότι η αρχή των συντεταγμένων είναι το σημείο (0,0,0).

β) Υπολογίστε με ακρίβεια 1‰, τη ροπή αδράνειας σφαίρας μοναδιαίας ακτίνας και πυκνότητας $\rho=1+\exp(r)$ ως προς άξονα που περνά από το κέντρο της σφαίρας.