

## Η μέθοδος του Verlet - εισαγωγικά

- Η μέθοδος του Euler στηρίζεται στον ορισμό της δεξιάς παραγώγου.  
Ένας ισοδύναμος ορισμός είναι

$$f'(t) = \lim_{h \rightarrow 0} \frac{f(t+h) - f(t-h)}{2h}$$

- Αυτή η εξίσωση λέμε ότι είναι ζυγισμένη ως προς  $t$ . Χρησιμοποιώντας το ανάπτυγμα Taylor μπορούμε να γράψουμε

$$f(t+h) = f(t) + hf'(t) + \frac{1}{2!}h^2 f''(t) + \frac{1}{3!}h^3 f'''(\zeta_+)$$

$$f(t-h) = f(t) - hf'(t) + \frac{1}{2!}h^2 f''(t) - \frac{1}{3!}h^3 f'''(\zeta_-)$$

όπου  $\zeta_{\pm}$  είναι μεταξύ  $t$  και  $t \pm h$

- Επομένως αφαιρώντας γράφουμε:

$$f'(t) = \frac{f(t+h) - f(t-h)}{2h} - \frac{1}{3!}h^2 f'''(\zeta) \quad \text{όπου } t-h \leq \zeta \leq t+h$$

σφάλμα αποκοπής τάξης  $h^2$

- Χρησιμοποιώντας το ανάπτυγμα Taylor για  $f(t+h)$  και  $f(t-h)$  μπορούμε να γράψουμε τη δεύτερη παράγωγο της  $f$  με τη μορφή

$$f''(t) = \frac{f(t+h) + f(t-h) - 2f(t)}{h^2} - \frac{1}{4!}h^2 f^{(4)}(\zeta) \quad \text{όπου } t-h \leq \zeta \leq t+h$$

# Εύρεση της 1<sup>ης</sup> και 2<sup>ης</sup> παραγώγου συνάρτησης



```

C=====
      Program DERIVATIVES
C=====
C Evresi tw'n paragogen tis f(x)=sin(x)
C xrisimipoivntas tis ekfraseis gia tin
C proti kai deyteri paragogo apo ti
C proigoymeni selida
C F1 = f', F2=f''
C D1 = sfalma stin F1
C D2 = sfalma stin F2
C=====
      Parameter(N=101)
      REAL X(N), F(N), F1(N)
      REAL D1(N),F2(N),D2(N)
      pi = 4.0*atan(1.0)
      h = pi/(2.*100.)
      Do 100 i =1, N
        x(I) = h*(i-1)
        F(I) = sin(x(i))
100 Continue
      CALL THREE(N,h,F,F1,F2)
      DO 300 I = 1, N
        D1(I) = F1(I) - COS(X(I))
        D2(I) = F2(I) + SIN(X(I))
        WRITE(6,99)X(I),F1(I),D1(I),F2(I),D2(I)
300 Continue
      99 FORMAT(5F10.6)
      END

```

```

C=====
      Subroutine THREE(N,H,FI,F1,F2)
C=====
C Yporoutina gia ton ypologismo tis
C prwtis kai deyteris paragogo.
C Input arguments:
C   h: vima
C   FI: input f(x)
C Output arguments:
C   F1 kai F2: 1η kai 2η paragogos
C=====
      REAL FI(N),F1(N),F2(N)
      DO 100 I =2, N-1
        F1(I) = (FI(I+1)-FI(I-1))/(2.0*h)
        F2(I) = (FI(I+1)-2.*FI(I)+FI(I-1))/(h*h)
100 Continue
      F1(1) = 2.0*F1(2) - F1(3)
      F1(N) = 2.0*F1(N - 1) - F1(N - 2)
      F2(1) = 2.0*F2(2) - F2(3)
      F2(N) = 2.0*F2(N - 1) - F2(N - 2)
      RETURN
      END

```

## Η Μέθοδος Verlet

- Παίρνοντας τις εξισώσεις κίνησης  $\frac{d\vec{r}}{dt} = \vec{v}(t); \quad \frac{d^2\vec{r}}{dt^2} = \vec{a}(\vec{r})$  και χρησιμοποιώντας τις σχέσεις για την πρώτη και δεύτερη παράγωγο που δείξαμε πριν θα έχουμε

$$\frac{\vec{r}_{n+1} - \vec{r}_{n-1}}{2h} + O(h^2) = \vec{v}_n \quad \frac{\vec{r}_{n+1} + \vec{r}_{n-1} - 2\vec{r}_n}{h^2} + O(h^2) = \vec{a}_n \quad \text{οπου} \quad \vec{a}_n = \vec{a}(\vec{r}_n)$$

- Από τις εξισώσεις αυτές παίρνουμε:

$$\vec{v}_n = \frac{\vec{r}_{n+1} - \vec{r}_{n-1}}{2h} + O(h^2)$$

$$\vec{r}_{n+1} = 2\vec{r}_n - \vec{r}_{n-1} + h^2\vec{a}_n + O(h^4)$$

- Γνωρίζοντας  $r_0$  και  $r_1$  μπορούμε να υπολογίσουμε  $r_2$ .  
Γνωρίζοντας  $r_2$  και  $r_1$  μπορούμε να υπολογίσουμε  $r_3$  και άρα να πάρουμε  $v_2$
- Το πρόβλημα της μεθόδου είναι ότι δεν είναι εύκολο να την ξεκινήσουμε.  
Οι αρχικές συνθήκες που μας δίνονται συνήθως είναι  $r_1 = r(t=0)$  και  $v_1 = v(t=0)$  αλλά όχι  $r_0 = r(t=-h)$
- Επιπλέον προβλήματα στρογγυλοποίησης στον τύπο της ταχύτητας λόγω αφαίρεσης σχεδόν ίσων και μεγάλων μεγεθών

## Η Μέθοδος Verlet – ταχύτητας

Από τις σχέσεις:  $\vec{v}_n = \frac{\vec{r}_{n+1} - \vec{r}_{n-1}}{2h} + O(h^2)$  (1)

$$\vec{r}_{n+1} = 2\vec{r}_n - \vec{r}_{n-1} + h^2\vec{a}_n + O(h^4) \quad (2)$$

Λύνοντας την (1) ως προς  $\vec{r}_{n-1}$  έχουμε:

$$\vec{r}_{n-1} = \vec{r}_{n+1} - 2h\vec{v}_n \quad (3)$$

Αντικαθιστώντας στην (2):

$$\vec{r}_{n+1} = \vec{r}_n + h\vec{v}_n + \frac{h^2}{2}\vec{a}_n \quad (4)$$

Από την (1) έχουμε ακόμα:

$$\vec{v}_{n+1} = \frac{\vec{r}_{n+2} - \vec{r}_n}{2h} \quad (5)$$

Αλλά σύμφωνα με την (2),  $\vec{r}_{n+2}$  δίνεται από:

$$\vec{r}_{n+2} = 2\vec{r}_{n+1} - \vec{r}_n + h^2\vec{a}_{n+1} \quad (6)$$

Αντικατάσταση της (6) στη (5) δίνει:

$$\vec{v}_{n+1} = \frac{\vec{r}_{n+1} - \vec{r}_n}{h} + \frac{h}{2}\vec{a}_{n+1} \quad (7)$$

Χρησιμοποιώντας την (4) για αντικατάσταση του  $\vec{r}_{n+1} - \vec{r}_n$  στην (7) έχουμε:

$$\vec{v}_{n+1} = \frac{h\vec{v}_n + \frac{h^2}{2}\vec{a}_n}{h} + \frac{h}{2}\vec{a}_{n+1} \Rightarrow \vec{v}_{n+1} = \vec{v}_n + \frac{h}{2}(\vec{a}_n + \vec{a}_{n+1})$$

$$\vec{r}_{n+1} = \vec{r}_n + h\vec{v}_n + \frac{h^2}{2}\vec{a}_n$$

Μέθοδος Verlet ταχύτητας

- Η μέθοδος αυτή είναι μαθηματικά ισοδύναμη με την γενική μορφή του Verlet αλλά δεν παρουσιάζει προβλήματα σφαλμάτων στο γυλοποίησης και επίσης ξεκινά εύκολα από τις αρχικές συνθήκες και μόνο.

## Η Μέθοδος Verlet – ταχύτητας

Σχηματικά η μέθοδος ταχύτητας Verlet μπορεί να χρησιμοποιηθεί ως εξής:

Δεδομένων των αρχικών συνθηκών  $x_k$  και  $v_k$  και της συνάρτησης της δύναμης ή επιτάχυνσης ενός σώματος  $F(x)$ :

**Βήμα 1:** Υπολογίζουμε 
$$v_{k+1/2} = v_k + \frac{h}{2} F(x_k)$$

**Βήμα 2:** Υπολογίζουμε 
$$x_{k+1} = x_k + v_{k+1/2} h$$

**Βήμα 3:** Υπολογίζουμε 
$$v_{k+1} = v_{k+1/2} + \frac{h}{2m} F(x_{k+1})$$

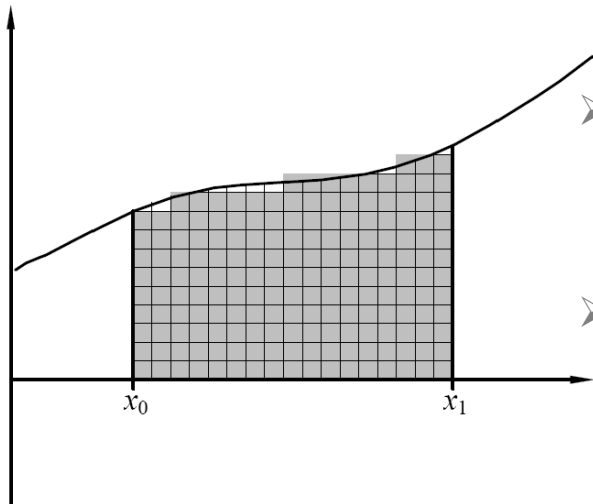
# Αριθμητική ολοκλήρωση

□ Υπάρχουν πολλοί λόγοι που κάποιος θέλει να κάνει αριθμητική ολοκλήρωση:

- Το ολοκλήρωμα είναι δύσκολο να υπολογισθεί αναλυτικά
- Ολοκλήρωση πίνακα δεδομένων

□ Διάφοροι τρόποι ολοκλήρωσης ανάλογα με το πρόβλημα

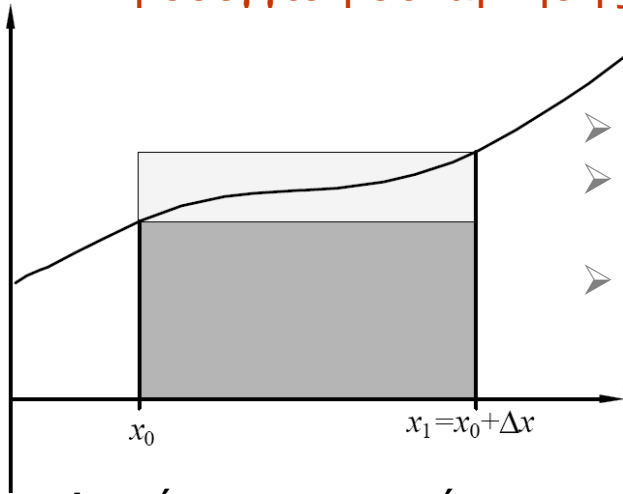
❖ Ολοκλήρωση με το “χέρι”



- Ένας τρόπος είναι να χρησιμοποιήσουμε ένα πλέγμα πάνω στο γράφημα της συνάρτησης προς ολοκλήρωση και να μετρήσουμε τα τετράγωνα (μόνο αυτά που περιέχονται κατά 50% από τη συνάρτηση).
- Αν οι υποδιαιρέσεις του πλέγματος (τετράγωνα) είναι πολύ μικρές τότε μπορούμε να προσεγγίσουμε αρκετά καλά το ολοκλήρωμα της συνάρτησης.

# Αριθμητική ολοκλήρωση

## ❖ Προσέγγιση συνάρτησης με σταθερά



- Ο πιο απλός τρόπος ολοκλήρωσης.
- Υποθέτουμε ότι η συνάρτηση  $f(x)$  είναι σταθερή στο διάστημα  $(x_0, x_1)$
- Η μέθοδος δεν είναι ακριβής και οδηγεί σε αμφίβολα αποτελέσματα ανάλογα με το αν η σταθερά επιλέγεται στην αρχή ή το τέλος του διαστήματος ολοκλήρωσης.

Αν πάρουμε το ανάπτυγμα Taylor της συνάρτησης  $f(x)$  ως προς το κατώτερο όριο:

$$\begin{aligned} \int_{x_0}^{x_0+\Delta x} f(x)dx &= \int_{x_0}^{x_0+\Delta x} \left[ f(x_0) + f'(x_0)(x-x_0) + \frac{1}{2}f''(x_0)(x-x_0)^2 + \dots \right] dx \\ &= f(x_0)\Delta x + \frac{1}{2}f'(x_0)(x-x_0)^2 + \frac{1}{6}f''(x_0)(x-x_0)^3 + \dots = \underbrace{f(x_0)}_{\text{σταθερά}} \Delta x + O(\Delta x^2) \end{aligned}$$

Αν η σταθερά λαμβάνεται από το πάνω όριο ολοκλήρωσης θα είχαμε:

$$\int_{x_0}^{x_0+\Delta x} f(x)dx = f(x_0 + \Delta x)\Delta x + O(\Delta x^2)$$

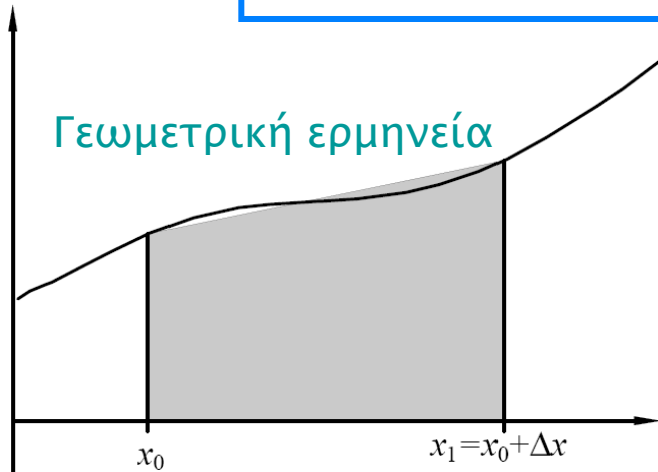
- Το σφάλμα και στις 2 περιπτώσεις είναι τάξης  $O(\Delta x^2)$  με το συντελεστή να καθορίζεται από τη τιμή της 1<sup>ης</sup> παραγώγου

# Αριθμητική ολοκλήρωση – Κανόνας του τραπεζίου

Θεωρήστε το ανάπτυγμα της σειράς Taylor που ολοκληρώνεται μεταξύ  $x_0$  και  $x_0 + \Delta x$

$$\begin{aligned}
 \int_{x_0}^{x_0 + \Delta x} f(x) dx &= \int_{x_0}^{x_0 + \Delta x} \left[ f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2} f''(x_0)(x - x_0)^2 + \dots \right] dx \\
 &= f(x_0)\Delta x + \frac{1}{2} f'(x_0)\Delta x^2 + \frac{1}{6} f''(x_0)\Delta x^3 + \dots \\
 &= \left\{ \frac{1}{2} f(x_0) + \frac{1}{2} \left[ f(x_0) + f'(x_0)\Delta x + \frac{1}{2} f''(x_0)\Delta x^2 + \dots \right] - \frac{1}{12} f''(x_0)\Delta x^2 + \dots \right\} \Delta x \\
 &= \boxed{\frac{1}{2} [f(x_0) + f(x_0 + \Delta x)] \Delta x} + \underbrace{O(\Delta x^3)}_{\text{Κανόνας του τραπεζίου}}
 \end{aligned}$$

Σφάλμα προσέγγισης

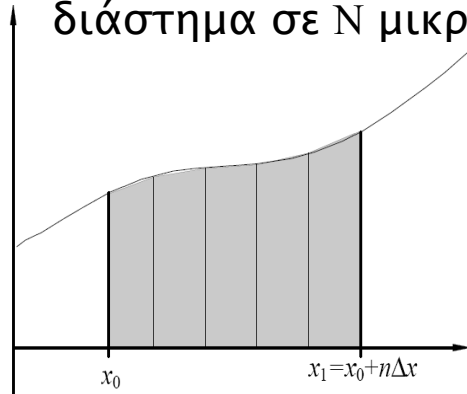


Αφού το σφάλμα ελαττώνεται κατά  $\Delta x^3$  κάνοντας το διάστημα μισό το σφάλμα θα μικραίνει κατά 8. Αλλά η περιοχή θα ελαττωθεί στο μισό και θα πρέπει να χρησιμοποιήσουμε το κανόνα 2 φορές και να αθροίσουμε. Το σφάλμα τελικά ελαττώνεται κατά 4.



# Αριθμητική ολοκλήρωση – Κανόνας του τραπεζίου

Συνήθως όταν θέλουμε να ολοκληρώσουμε σε ένα διάστημα  $x_0, x_1$  χωρίζουμε το διάστημα σε  $N$  μικρότερα διαστήματα  $\Delta x = (x_1 - x_0)/N$



Εφαρμόζοντας το κανόνα του τραπεζίου

$$\int_{x_0}^{x_1} f(x) dx = \sum_{i=0}^{n-1} \int_{x_0 + i\Delta x}^{x_0 + (i+1)\Delta x} f(x) dx$$

$$\approx \frac{\Delta x}{2} \sum_{i=0}^{n-1} [f(x_0 + i\Delta x) + f(x_0 + (i+1)\Delta x)] \Rightarrow$$

$$\int_{x_0}^{x_1} f(x) dx \approx \frac{\Delta x}{2} [f(x_0) + 2f(x_0 + \Delta x) + 2f(x_0 + 2\Delta x) + \dots + 2f(x_0 + (n-1)\Delta x) + f(x_1)]$$

Ενώ το σφάλμα για κάθε βήμα είναι  $\Delta x^3$  το συνολικό σφάλμα είναι αθροιστικό ως προς όλα τα βήματα ( $N$ ) και επομένως θα είναι  $N$  φορές  $O(\Delta x^2) \sim O(N^{-2})$

Στα παραπάνω υποθέσαμε ότι το βήμα,  $\Delta x$ , είναι σταθερό σε όλο το διάστημα. Θα μπορούσε ωστόσο να μεταβάλλεται σε μια περιοχή (να 'ναι πιο μικρό) ώστε να έχουμε μικρότερο σφάλμα. π.χ. περιοχές με μεγάλη καμπύλωση της συνάρτησης **Προσοχή** το σφάλμα παραμένει και πάλι της τάξης  $O(\Delta x^2)$  αλλά οι υπολογισμοί θα είναι πιο ακριβείς

Αυτό το κάνουμε γιατί σε περιοχές μεγάλης καμπύλωσης η 2<sup>η</sup> παράγωγος της συνάρτησης θα γίνει πολύ μεγάλη οπότε μικρότερο  $\Delta x$  θα κρατήσει τον όρο μικρό



## Κώδικας για κανόνα του τραπεζίου

```

PROGRAM Integration
Double Precision x0,x1,Value,Exact,pi
INTEGER i, j, nx
C=====Functions
Double Precision TrapeziumRule
C=====Constants
pi = 2.0*ASIN(1.0D0)
Exact = 2.0
C=====Limits
x0 = 0.0
x1 = pi
C=====
C= Trapezium rule =
C=====
WRITE(6,*)
WRITE(6,*)'Trapezium rule'
nx = 1
DO i=1,20
  Value = TrapeziumRule(x0,x1,nx)
  WRITE(6,*)nx,Value,Value - Exact
  nx = 2*nx
ENDDO

```

### FUNCTION MYFUNC(x)

```

Double Precision x, myfunc
myfunc = SIN(x)
RETURN
END

```

### Double Precision FUNCTION TrapeziumRule(x0,x1,nx)

```

C=====parameters INTEGER nx
Double Precision x0,x1
C=====functions
Double precision myfunc
C=====local variables
INTEGER i
Double Precision dx, xa, xb, fa, fb, Sum
dx = (x1 - x0)/DFLOAT(nx)
Sum = 0.0
DO i=0,nx-1
  xa = x0 + DFLOAT(i)*dx
  xb = x0 + DFLOAT(i+1)*dx
  fa = myfunc(xa)
  fb = myfunc(xb)
  Sum = Sum + fa + fb
ENDDO
Sum = Sum * dx / 2.0
TrapeziumRule = Sum
RETURN
END

```

# Αριθμητική ολοκλήρωση – Κανόνας μέσου σημείου

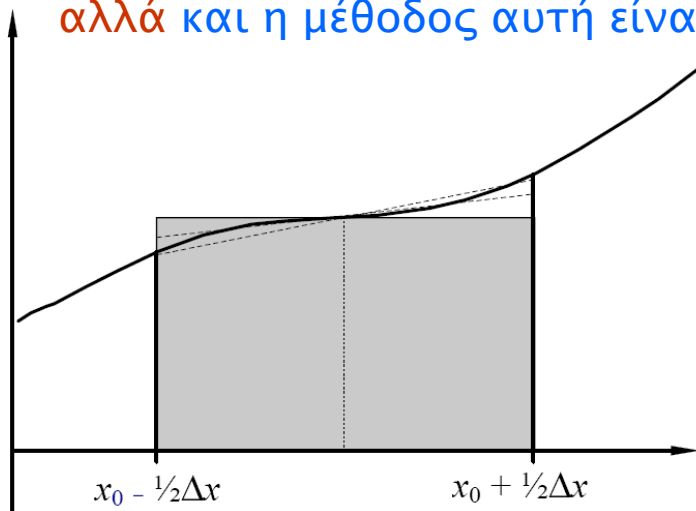
Μια παραλλαγή του κανόνα του τραπεζίου είναι ο κανόνας του μέσου σημείου

Η ολοκλήρωση του αναπτύγματος Taylor γίνεται από  $x_0 - \Delta x/2$  σε  $x_0 + \Delta x/2$  οπότε:

$$\int_{x_0 - \frac{1}{2}\Delta x}^{x_0 + \frac{1}{2}\Delta x} f(x) dx = \int_{x_0 - \frac{1}{2}\Delta x}^{x_0 + \frac{1}{2}\Delta x} \left[ f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2} f''(x_0)(x - x_0)^2 + \dots \right] dx \Rightarrow$$

$$\int_{x_0 - \frac{1}{2}\Delta x}^{x_0 + \frac{1}{2}\Delta x} f(x) dx \approx f(x_0)\Delta x + \frac{1}{24} f''(x_0)\Delta x^3 + \dots$$

Υπολογίζοντας τη συνάρτηση στο **μέσο κάθε διαστήματος** το σφάλμα μπορεί να ελαττωθεί κάπως μια και ο παράγοντας μπροστά από τον όρο της 2<sup>ης</sup> παραγώγου είναι 1/24 αντί του 1/12 που έχουμε στη μέθοδο του τραπεζίου **αλλά και η μέθοδος αυτή είναι τάξης  $O(\Delta x^2)$**



Διαχωρίζοντας το διάστημα σε υποδιαστήματα μπορούμε να ελαττώσουμε το σφάλμα όπως και στην περίπτωση του κανόνα του τραπεζίου:

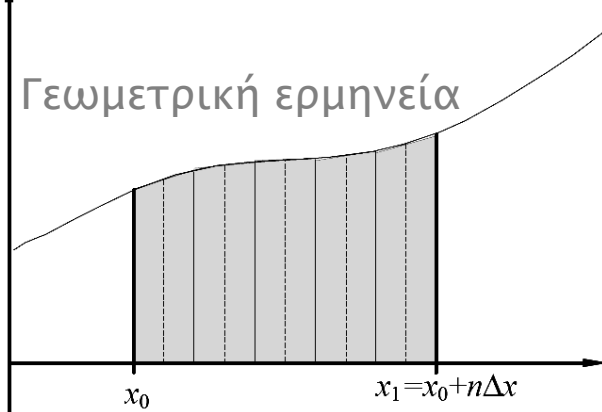
$$\int_{x_0}^{x_1} f(x) dx \approx \Delta x \sum_{i=0}^{n-1} f\left(x_0 + \left(i + \frac{1}{2}\right)\Delta x\right)$$

# Αριθμητική ολοκλήρωση – Κανόνας μέσου σημείου

Κανόνας μέσου σημείου: 
$$\int_{x_0}^{x_1} f(x)dx \approx \Delta x \sum_{i=0}^{n-1} f\left(x_0 + \left(i + \frac{1}{2}\right)\Delta x\right)$$

Κανόνας τραπεζίου: 
$$\int_{x_0}^{x_1} f(x)dx \approx \frac{\Delta x}{2} \left[ f(x_0) + 2f(x_0 + \Delta x) + 2f(x_0 + 2\Delta x) + \dots + 2f(x_0 + (n-1)\Delta x) + f(x_1) \right]$$

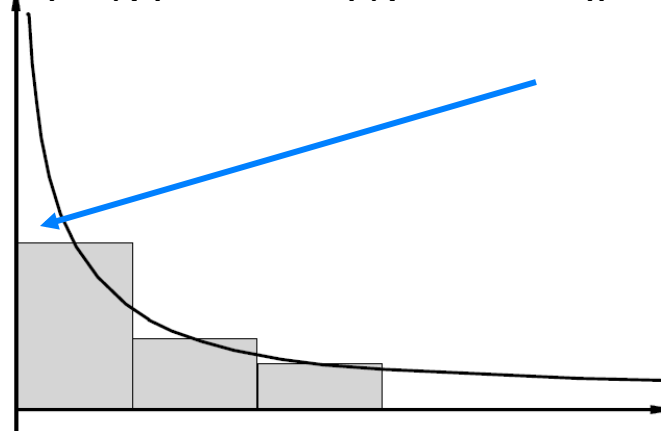
Η διαφορά τους είναι μόνο στη “φάση” των σημείων και της περιοχής υπολογισμού, και στον τρόπο υπολογισμού του πρώτου και τελευταίου διαστήματος



Ωστόσο υπάρχουν **δύο πλεονεκτήματα**

της μεθόδου του ενδιάμεσου σημείου σε σχέση με την μέθοδο του τραπεζίου:

- A) Χρειάζεται ένα υπολογισμό της  $f(x)$  λιγότερο
- B) Μπορεί να χρησιμοποιηθεί πιο αποτελεσματικά για τον υπολογισμό του ολοκληρώματος κοντά σε μια περιοχή που υπάρχει ολοκληρώσιμο ιδιάζων σημείο



# Κώδικας για κανόνα του ενδιάμεσου σημείου



```

PROGRAM Integration
Double Precision x0,x1,Value,Exact,pi
INTEGER i, j, nx
C=====Functions
Double Precision MidPointRule
C=====Constants
pi = 2.0*ASIN(1.0D0)
Exact = 2.0
C=====Limits
x0 = 0.0
x1 = pi
C=====
C= Mid point =
C=====
WRITE(6,*)
WRITE(6,*)'Midpoint rule'
nx = 1
DO i=1,20
  Value = MidPointRule(x0,x1,nx)
  WRITE(6,*)nx,Value,Value – Exact
  nx = 2*nx
ENDDO

```

```

FUNCTION myfunc(x)
Double Precision x, myfunc
myfunc = SIN(x)
RETURN
END

```

```

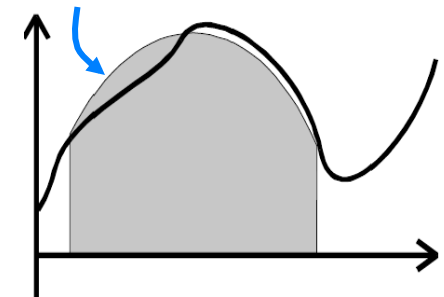
Double Precision FUNCTION MidpointRule(x0,x1,nx)
C=====parameters
INTEGER nx
Double Precision x0,x1
C=====functions
Double Precision myfunc
C=====local variables
INTEGER i
Double Precision dx,xa,fa,Sum
dx = (x1 - x0)/Dfloat(nx)
Sum = 0.0
DO i=0,nx-1
  xa = x0 + (DFLOAT(i)+0.5)*dx
  fa = myfunc(xa)
  Sum = Sum + fa
ENDDO
Sum = Sum * dx
MidpointRule = Sum
RETURN
END

```

# Αριθμητική ολοκλήρωση – Κανόνας του Simpson

Ένας διαφορετικός τρόπος προσέγγισης από το να ελαττώνουμε το μέγεθος του  $\Delta x$  στην ολοκλήρωση για **καλύτερη ακρίβεια** είναι να **αυξήσουμε την ακρίβεια των συναρτήσεων που χρησιμοποιούμε για να προσεγγίσουμε το ολοκλήρωμα**.

2<sup>ου</sup> βαθμού προσέγγιση της  $f(x)$



Ολοκληρώνοντας το ανάπτυγμα Taylor σε ένα διάστημα  $2\Delta x$

$$\int_{x_0}^{x_0+2\Delta x} f(x)dx = 2f(x_0)\Delta x + 2f'(x_0)\Delta x^2 + \frac{4}{3}f''(x_0)\Delta x^3 + \frac{2}{3}f'''(x_0)\Delta x^4 + \frac{4}{15}f^{iv}(x_0)\Delta x^5 \dots$$

$$\begin{aligned} \int_{x_0}^{x_0+2\Delta x} f(x)dx &= \frac{\Delta x}{3} \left[ f(x_0) + \right. \\ &\quad \left. + 4 \left( f(x_0) + f'(x_0)\Delta x + \frac{1}{2}f''(x_0)\Delta x^2 + \frac{1}{6}f'''(x_0)\Delta x^3 + \frac{1}{24}f^{iv}(x_0)\Delta x^4 + \dots \right) \right. \\ &\quad \left. + \left( f(x_0) + 2f'(x_0)\Delta x + 2f''(x_0)\Delta x^2 + \frac{4}{3}f'''(x_0)\Delta x^3 + \frac{2}{3}f^{iv}(x_0)\Delta x^4 + \dots \right) - \right. \\ &\quad \left. - \frac{17}{30}f^{iv}(x_0)\Delta x^4 \dots \right] \Rightarrow \end{aligned}$$

$$\int_{x_0}^{x_0+2\Delta x} f(x)dx = \frac{\Delta x}{3} \left[ f(x_0) + 4f(x_0 + \Delta x) + f(x_0 + 2\Delta x) \right] + O(\Delta x^5)$$

Ο κανόνας του Simpson είναι 2 τάξεις περισσότερο ακριβής από αυτόν του τραπεζίου δίνοντας ακριβή ολοκλήρωση κύβων

## Αριθμητική ολοκλήρωση – Βελτιστοποίηση Simpson

Χωρίζοντας το διάστημα από  $x_0$  σε  $x_1$  σε μικρότερα ( $N$ ) διαστήματα, μπορούμε να αυξήσουμε την ακρίβεια του αλγόριθμου.

Το γεγονός ότι χρειαζόμαστε 3 σημεία για την ολοκλήρωση κάθε διαστήματος απαιτεί να υπάρχουν **άρτια** σε πλήθος διαστήματα.

Άρα θα πρέπει να εκφράσουμε το πλήθος των διαστημάτων σαν  $N = 2m$ .

Θα έχουμε:

$$\int_{x_0}^{x_1} f(x)dx \approx \frac{\Delta x}{3} \sum_{i=0}^{m-1} \left[ f(x_0 + 2i\Delta x) + 4f(x_0 + (2i+1)\Delta x) + f(x_0 + (2i+2)\Delta x) \right]$$

$$\int_{x_0}^{x_1} f(x)dx \approx \frac{\Delta x}{3} \left[ f(x_0) + 4f(x_0 + \Delta x) + 2f(x_0 + 2\Delta x) + \dots + 4f(x_0 + (N-1)\Delta x) + f(x_1) \right]$$

# Κώδικας για κανόνα του Simpson



```

PROGRAM Integration
Double Precision x0,x1,Value,Exact,pi
INTEGER i, j, nx
C=====Functions
Double Precision SimpsonsRule
C=====Constants
pi = 2.0*ASIN(1.0D0)
Exact = 2.0
C=====Limits
x0 = 0.0
x1 = pi
C=====
C= Simpson =
C=====
WRITE(6,*)
WRITE(6,*)'Simpson ' ' rule'
nx = 2
DO i=1,10
  Value = SimpsonsRule(x0,x1,nx)
  WRITE(6,*) nx,Value,Value – Exact
  nx = 2*nx
ENDDO

```

```

FUNCTION myfunc(x)
Double Precision x, myfunc
myfunc = SIN(x)
RETURN
END

```

```

Double Precision FUNCTION SimpsonsRule(x0,x1,nx)
INTEGER nx
Double Precision x0,x1
Double Precision myfunc
INTEGER i
Double Precision dx,xa,xb,xc,fa,fb,fc,Sum
dx = (x1 - x0)/DFLOAT(nx)
Sum = 0.0
DO i=0, nx-1, 2
  xa = x0 + DFLOAT(i)*dx
  xb = x0 + DFLOAT(i+1)*dx
  xc = x0 + DFLOAT(i+2)*dx
  fa = myfunc(xa)
  fb = myfunc(xb)
  fc = myfunc(xc)
  Sum = Sum + fa + 4.0*fb + fc
ENDDO
Sum = Sum * dx / 3.0
SimpsonsRule = Sum
RETURN
END

```



## Ολοκλήρωση - Πολλαπλά ολοκληρώματα

Έστω ότι έχουμε να υπολογίσουμε ένα ολοκλήρωμα της μορφής:

$$\int_{x_a}^{x_b} \int_{y_a}^{y_b} f(x,y) dx dy$$

$$f(x) = \int_{y_a}^{y_b} f(x,y) dy$$

$$\int_{x_a}^{x_b} f(x) dx$$

Μπορεί να λυθεί υπολογίζοντας το καθένα ολοκλήρωμα κατά σειρά

Κανόνα του τραπεζίου

Κανόνα ενδιάμεσου σημείου

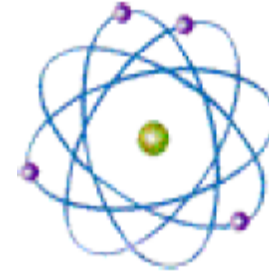
Κανόνα Simpson

Θεωρώντας  $N$  σημεία για κάθε ολοκλήρωση θα χρειαστούμε  $N^2$  υπολογισμούς

## Ολοκλήρωση - Πολλαπλά ολοκληρώματα

Για παράδειγμα, έστω τα ηλεκτρόνια του ατόμου του  ${}^4\text{Be}$ :

$$I = \int_0^1 dx_1 \int_0^1 dx_2 \int_0^1 dx_3 \cdots \int_0^1 dx_{12} f(x_1, x_2, \dots, x_{12})$$



Έχουμε 3 διαστάσεις για κάθε ηλεκτρόνιο x 4 ηλεκτρόνια = 12 διαστάσεις

Αλλά για 100 σημεία για κάθε ολοκλήρωση τότε θα χρειαστούμε συνολικά

$$100^{12} = 10^{24} \text{ υπολογισμούς}$$

Για τα PCs υποθέτοντας 1Giga υπολογισμούς/sec θα χρειαστούμε  $10^7$  χρόνια

# Ολοκλήρωση - Μέθοδος Monte Carlo

Χρησιμοποίηση τυχαίων αριθμών για επίλυση ολοκληρωμάτων

Η μέθοδος Monte Carlo δίνει μια διαφορετική προσέγγιση για την επίλυση ενός ολοκληρώματος

## Τυχαίοι αριθμοί

Η συνάρτηση `rand` προσφέρει μια ακολουθία τυχαίων αριθμών ομοιόμορφα κατανεμημένων στο διάστημα  $[0,1)$

Δύο βασικές μέθοδοι χρησιμοποιούνται για την επίλυση ολοκληρωμάτων

Μέθοδος επιλογής ή δειγματοληψίας

Μέθοδος μέσης τιμής

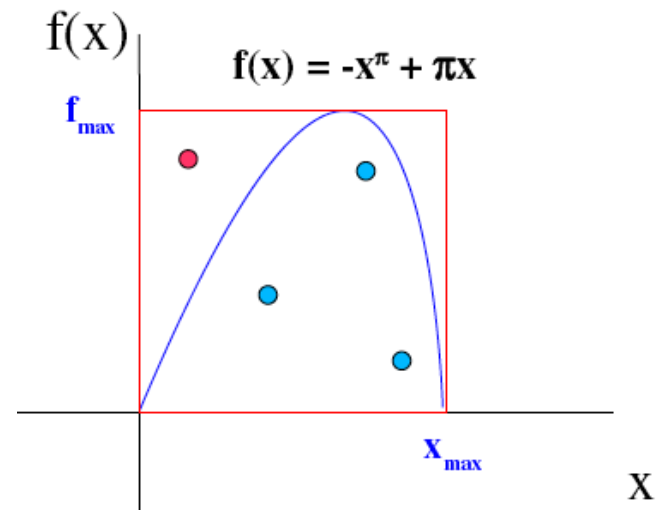
# Monte Carlo μέθοδος δειγματοληψίας

- Περικλείουμε την συνάρτηση που θέλουμε να ολοκληρώσουμε μέσα σε ένα ορθογώνιο στο διάστημα της ολοκλήρωσης
  - ❑ Υπολογίζουμε το εμβαδό του ορθογωνίου
- Εισάγουμε τυχαία σημεία στο ορθογώνιο
- Μετρούμε τα σημεία που βρίσκονται μέσα στο ορθογώνιο και αυτά που περικλείονται από την συνάρτηση
- Το εμβαδό της συνάρτησης (ολοκλήρωμα) στο διάστημα ολοκλήρωσης δίνεται από

$$E_{f(x)} = E_{\text{ορθογ.}} \times \frac{N_{f(x)}}{N_{\text{ορθογ.}}}$$

Όπου  $N_{f(x)}$  = αριθμός ●

$N_{\text{ορθογ.}}$  = αριθμός ●

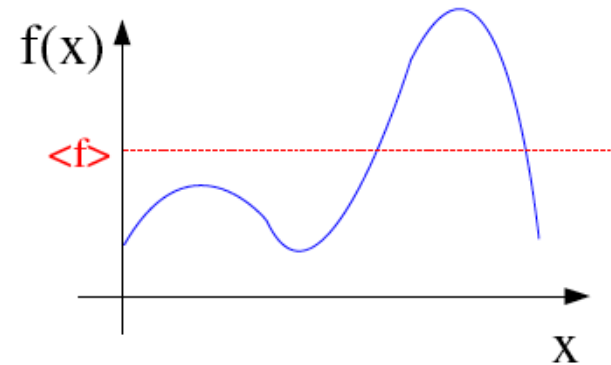


## Monte Carlo μέθοδος μέσης τιμής

Η ολοκλήρωση με Monte Carlo γίνεται με το να πάρουμε τη μέση τιμή της συνάρτησης υπολογιζόμενη σε τυχαία επιλεγμένα σημεία μέσα στο διάστημα ολοκλήρωσης

$$I = \int_{x_a}^{x_b} f(x) dx = (b - a) \langle f(x) \rangle$$

$$\langle f(x) \rangle \approx \frac{1}{N} \sum_{i=0}^N f(x_i)$$



Το στατιστικό σφάλμα:  $\delta I = \sigma_{\bar{f}}$  όπου  $\sigma_{\bar{f}} = \frac{\sigma_f}{\sqrt{N}}$

# Παράδειγμα κώδικα ολοκλήρωσης Monte Carlo

```

program random
integer iseed/12345/
call srand(iseed)      ! Ξεκίνημα ακολουθίας τυχαίων αριθμών με
                        ! αρχική τιμή 12345

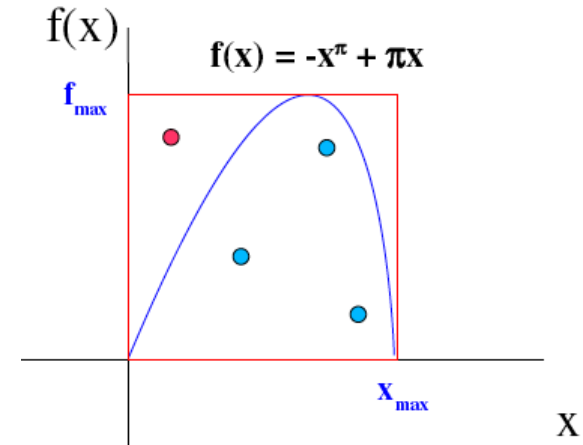
```

## Τμήμα κώδικα για τη Μέθοδο δειγματοληψίας

```

Do I = 0, Npnts
  x = xmax * rand() ! Η συνάρτηση rand() επιστρέφει ψευδοτυχαίες
                    ! τιμές στο διάστημα [0,1)
  fRand = fMax * rand()
  if (fRand < myfunc(x)) then
    below = below + 1
  endif
Enddo
Print *, ' Apotelesma oloklirwsis = ', fmax*xMax *below/Npnts

```

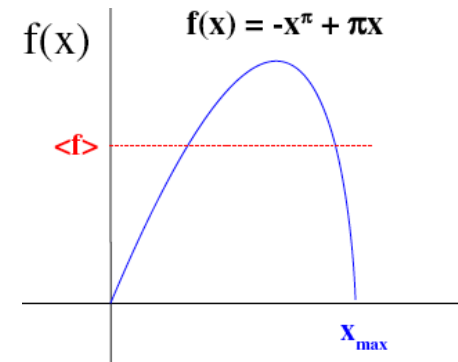


## Τμήμα κώδικα για τη Μέθοδο μέσης τιμής

```

Sum = 0
Do I = 0, Npnts
  x = xmax * rand()
  sum = f(x) + sum
End do
Print *, ' Apotelesma oloklirwsis ', xmax*sum/Npnts

```



## Monte Carlo ολοκλήρωση σε πολλές διαστάσεις

Εύκολο να γενικεύσουμε τη μέθοδο της μέσης τιμής σε πολλές διαστάσεις  
Το σφάλμα στη μέθοδο ολοκλήρωσης με Monte Carlo είναι στατιστικό

Ελαττώνεται ως  $1/\sqrt{N}$

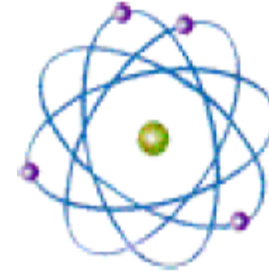
Για 2 διαστάσεις:

$$I = \int_a^b dx_1 \int_c^d dx_2 f(x, y) \simeq (b-a)(d-c) \times \frac{1}{N} \sum_i^N f(x_i, y_i)$$

## Ολοκλήρωση - Πολλαπλά ολοκληρώματα

Για παράδειγμα, έστω τα ηλεκτρόνια του ατόμου του  ${}^4\text{Be}$ :

$$I = \int_0^1 dx_1 \int_0^1 dx_2 \int_0^1 dx_3 \cdots \int_0^1 dx_{12} f(x_1, x_2, \dots, x_{12})$$



Έχουμε 3 διαστάσεις για κάθε ηλεκτρόνιο x 4 ηλεκτρόνια = 12 διαστάσεις

$$I = (1-0)^{12} \times \frac{1}{N} \sum f(x_1^i, x_2^i, \dots, x_{12}^i)$$

Αλλά για  $N=10^6$  σημεία στη ολοκλήρωση Monte Carlo έχουμε  $10^6$  υπολογισμούς

Για τα PCs υποθέτοντας 1Giga υπολογισμούς/sec θα χρειαστούμε  $10^{-3}$  secs !!