هندسة / ثاني ميكانيك / برمجة

الفصل الأول
الثوابت والمتغيرات

1.1 مقدمة

أشتقت كلمة فورتران Fortran من المقطعين الأولين من الكلمتين Formula Translation أو الكلمتين Formula Translator وتعني مترجم المعادلات أو ترجمة المعادلات. وقد أستحدثت لغة فورتران عام ١٩٥٧ لغرض برمجة التطبيقات في المجالات العلمية. تعتبر لغة فورتران الكثر اللغات الراقية شيوعاً وأستعمالاً وذلك لأهميتها وسهولة تعلمها ومناسبتها لحل المشكلات الرياضية والعلمية. وقد أجري عليها تعديلات كثيرة نتيجة للخبرات التي أكتسبها العلماء والمبرمجون على مر السنين.

ففي عام ١٩٧٧ تم أصدار صورة من صور فورتران وقد سميت فورتران ٧٧ والتي هي موضوع دراستنا. فكانت أخر صورة من صور فورتران هي فورتران ٩٠، والتي وضعت أصولها في عام ١٩٩٠.

1-2 الرموز الاساسية في لغة فورتران Fortran Symbols

أن الرموز المستخدمة في لغة فورتران هي:

a - الارقام الحسابية وهي (٠, ١, ... ٩).

b - الحروف الانكليزية وهي (A - Z).

c - الرموز الخاصة وتشمل ما يأتي:

$	أشارة دولار Dollar Sign
b	فراغ Blank (Space)
.	فاصلة عشرية Decimal Point
,	فاصلة عادية Comma
=	أشارة مساواة Equal Sign

Left paranthesis	(قوس أيسر
Right paran thesis	) قوس أيمن
plus sign	+ أشارة زائد
Minus sign	- أشارة ناقص
Asterisk	* نجمة
Double Asterisk	** نجمتان
Slash	/ شرطة
Quotation Mark	" حاصرة عليا
Colon	: نقطتا تفسير

1.3 الثوابت Constants

وهي القيم التي تبقى ثابتة خلال تنفيذ البرنامج وتقسم الى :-

1. الثوابت العددية Numerical Constants

وتنقسم الى :

a - الثوابت العددية الصحيحة Integer Constants

وتشمل الاعداد التي لا تحتوي على فاصلة عشرية مثل

31 100 -61 +47 0

b - الثوابت العددية الحقيقية Real Constants

وتشمل الاعداد التي تحتوي على فاصلة عشرية أو الكسر مثل

1.0 0.5 26.0 -1.63 +26.5

يمكن تمثيل الثوابت العددية بالصيغة الاسية خصوصاً اذا كانت قيمة الثوابت صغيرة أو كبيرة جداً ومثال ذلك

1.35E-3 = 1.35×10^{-3} = 0.00135

2.0E4 = 2.0×10^{4} = 20000.0

4E2 = 4×10^{2} = 400

أمثلة غير مقبولة للثوابت الصحيحة

31.3 تحتوي على فاصلة عشرية

1,0 تحتوي على فاصلة

50M لا يمثل عدداً مؤلفاً من أرقام فقط

أمثلة غير مقبولة للثوابت الحقيقية

1000 لا تحتوي على فاصلة عشرية

1,414.5 تحتوي على فارزة

A6.2 تحتوي على رمز

2- الثوابت المنطقية Logical Constants

وهي ثوابت يمكن تحديدها بأحد القيمتين الآتيتين:

قيمة حقيقية: TRUE قيمة زائفة: FALSE

3- الثوابت المركبة Complex Constants

الثابت المركب عبارة عن ثابت عددي ذو فاصلة عشرية وهو مكون من جزئين يسمى الجزء الأول بالجزء الحقيقي (Real part) ويسمى الجزء الثاني بالجزء التخيلي (Imaginary part)، فمثلاً 3.2-i1.86 يكتب بالصيغة التالية بلغة فورتران (1.86, 3.2).

4- الثوابت الرمزية Character Constants

تستخدم هذه الثوابت في كتابة العناوين وتميز بعض نتائج البرامج، وهذا النوع من الثوابت غير عددي أي لا تخضع للعمليات الحسابية ومثال ذلك:

'TABLE' 'QUALITY' 'AMOUNT'

1.4 المتغيرات Variables

أن المتغيرات تتألف من أسماء لمواقع في الذاكرة وأن محتويات هذه المواقع يمكن أن تتغير أثناء تنفيذ البرنامج، ومن هذه المتغيرات:

1- المتغيرات العددية الصحيحة Integer Variables

هي متغيرات تأخذ قيم ثوابت عددية صحيحة، وتخضع للقواعد الآتية:

a- يبدأ أسم المتغير بأحد الأحرف (I, J, K, L, M, N).
b- يتكون أسم المتغير من أحرف (A-Z) وأرقام حسابية (0-9).
c- لا يزيد أسم المتغير عن ستة رموز.

ومثال ذلك:

I2, KLM, NIGT, N2, LTGHT, JJ, NG

يمكن أستعمال متغيرات تبدأ بحروف أخرى غير المذكورة اعلاه على انها متغيرات صحيحة، ولكن يجب الاعلان عنها في بداية البرنامج وكما موضح في المثال التالي

INTEGER Y3, ZX, A

أما الأمثلة الآتية، فهي غير صحيحة، للأسباب الواردة أزاء كل منها:

3I يبدأ الأسم برقم وليس بحرف.
NAME1 يحتوي الأسم على رمز خاص (-).
IJKLMN عدد الرموز اكثر من ستة.
HELLO يبدأ الأسم بالحرف H، وهو ليس من الحروف الستة.

2- المتغيرات العددية الحقيقية Real Variables

هي متغيرات تأخذ قيم ثوابت عددية حقيقية، وتخضع لنفس قواعد المتغيرات الصحيحة ماعدا أن المتغير يبدأ بحروف أخرى غير المذكورة اعلاه ومثال ذلك:

FORCE, SYSTEM, X32, A, B6

الامثلة الاتية، غير صحيحة للاسباب الواردة ازاء كل منها:

JUMP الاسم يبدأ بالحرف J

X+Y يحتوي الاسم على رمز خاص (+)

4XY يبدأ الاسم برقم

يمكن استعمال متغيرات تبدأ بأحد الحروف (I, J, K, L, M, N) كمتغيرات حقيقية بشرط الاعلان عنها في بداية البرنامج وكما موضح في المثال التالي :-

REAL I, MN, J3

3- المتغيرات المركبة COMPLEX Variable

هي متغيرات تأخذ قيم ثوابت مركبة، ويتم الاعلان عن هذا النوع من المتغيرات كما موضح في المثال الاتي :-

COMPLEX BZ

BZ = (2, 3) BZ = 2+i3

4- المتغيرات المنطقية Logical Variable

متغيرات تأخذ قيمة منطقية أما صواب TRUE أو خطأ FALSE يجب الاعلان عن المتغيرات المنطقية في بداية البرنامج كما موضح في المثال التالي:

LOGICAL MN, PQ

5- المتغيرات الرمزية Character Variable

متغيرات تأخذ قيم ثوابت رمزية، ويجب الاعلان عن المتغير الرمزي وكم يحتوا من رموز في بداية البرنامج كما في المثال التالية

CHARACTER I* 10

أي انا اقصى عدد رموز مستقلة هي 10 رموز

CHARACTER *10 , I, J, K

اي ان اقصى عدد رموز مستقلة لكل من (I, J, K) هو 10 رموز.

5.1 التعبير الحسابي Arithmetic Expression

تجري العمليات الحسابية على كميات عددية والنتيجة تكون عددية ايضاً وتشمل العمليات التالية

العملية	رمز العملية	الاسبقية	مثال	
الرفع للقوة	**	1	A^B	A**B
الضرب	*	2	AB	A*B
القسمة	/	2	$\frac{A}{B}$	A/B
الجمع	+	3		A+B
الطرح	-	3		A-B

وتجري العمليات الحسابية وفقاً لسلم الاسبقيات المبين اعلاه مالم توضع أقواس لتحديد الاسبقية. أما اذا وضعت أقواس فتجري العمليات الحسابية ابتداءاً من اليسار كما في الامثلة الاتية:

5+6*7 = 47

(5+6)*(7) = 77

وتقسم التعابير الحسابية الى ثلاث اقسام:-

أ- التعبير الحسابي من النوع الصحيح

يتضمن ثوابت ومتغيرات عددية صحيحة، وتكون النتيجة ثابتاً عدداً صحيحاً ومثال ذلك:

2*I + 5

3/2 = 1

b- التعبير الحسابي من النوع الحقيقي
يتضمن ثوابت ومتغيرات عددية حقيقية وتكون النتيجة ثابتاً عددياً حقيقاً ومثاله :-

2.0 * X + 5.0
3.0 / 2.0 = 1.5

c- التعبير الحسابي من النوع المركب
يتضمن كلا النوعين السابقين من ثوابت أو متغيرات عددية وكما سوف في الأمثلة الآتية :-

3.2 + 18/5 = 6.2
3.0/2 = 1.5

1.6 الدوال الحسابية Mathematical Functions

الصيغة الرياضية	صيغته بلغة فورتران
$\sqrt{x}$	SQRT (X)
$\|x\|$	ABS (X)
e^x	EXP(X)
$\log(x)$	ALOG 10 (X)
$\ln(x)$	ALOG (X)
$\sin(x)$	SIN(X)
$\cos(x)$	COS(X)
$\tan(x)$	TAN(X)
$\cot(x)$	COTAN(X)
$\tan^{-1}(x)$	ATAN()
$\sinh(x)$	SINH(X)
$\cosh(x)$	COSH (X)

TANH(X)	tanh(x)
FLOAT (I)	تحويل المتغير الصحيح I الى متغير حقيقي في نظام اسرع الحاسوب
INT(X)	تحويل المتغير الحقيقي X الى متغير صحيح البرنامج

اذا كان التعبير الحسابي خالياً من الاقواس ومحتوياً على دوال حسابية فأن استخراج قيم هذه الدوال يبطل قاعدة الاسبقية لعمليات الرفع للقوة والضرب والقسمة، اي انه يسبق تلك العمليات.

1.7 رقم الجملة Statement Number

عند كتابة أي برنامج، فأننا نحتاج الى ترقيم بعض الجمل لتنفيذ بعض التعليمات، وعند الحاجة الى أي جملة فأنه يشار الى رقمها فقط. كما في المثال التالي:-

```
20  Y= X **2 + 5.0 * X
```

ومن الممكن اختيار أي رقم أخر لها على أن لا يزيد عن خمسة أرقام وفي لغة فورتران، لا يشترط أي ترتيب لأرقام الجمل في البرنامج من حيث تسلسلها.

1.8 جملة PARAMETER

تستخدم لتسمية الثوابت العددية في بداية البرنامج، ويمكن استخدام هذه الاسماء في أي مكان من البرنامج عند الحاجة اليها. ومن الامثلة على ذلك:-

```
PARAMETER ( PI=3.14159)
```

ويمكن تعريف اكثر من ثابت في سطر واحد

```
PARAMETER ( PI=3.14159 , E=2.718 )
```

مثال: اكتب التعابير الآتية بلغة فورتران

a- $\sqrt{x^2 - 4ay}$ = SQRT(X*X - 4.0*A*Y)

b- $\tan|a-b|/(a+b)$ = TAN(ABS(A-B))/(A+B)

c- $(\cos y)(\tan^{-1}(x/3))$ = COS(Y)*ATAN(X/3.0)

d- $X + \frac{Z}{\sqrt{k-L}}$ = X + Z/SQRT(FLOAT(K-L))

e- $\cosh(x+y)$ = COSH(X+Y)

وأنت

الفصل الثاني
عمل الادخال والاخراج

2.1 جمل الادخال INPUT STATEMENTS

جمل الادخال نوعان :

a - الجملة الحسابية :

تستخدم هذه الطريقة عند ادخال عدد قليل من المعطيات التي تستعمل لاجتداء البرنامج ومثال ذلك

A = 8.5
I = 9

b - جملة القراءة READ STATEMENTS

وهي على نوعين :

1 - جملة القراءة غير المصاغة :

وصيغتها العامة

READ*, List of Variables

ومثال ذلك

READ*, A, B, N

حيث تقوم بقراءة المتغيرات N, B, A على الترتيب، ويتم ادخال قيمها بواسطة شاشة الحاسبة، على أن تترك مسافات كافية بين هذه القيم حيث يمكن ان تمثل القيم الواردة في جملة القراءة

14.6 55.4 12

2 - جملة القراءة المصاغة :

صيغة الأولى

READ (n, m) List of Variables
m FORMAT (.....)

حيث :
n : رقم جهاز قراءة المعطيات للمتغيرات
m : رقم جملة الصيغة ، والتي توضح كيفية قراءة قيم المتغيرات .

```
READ (5,30) Y, L, X, Z
30 FORMAT (F6.2, I3, 2F4.1)
```

الصيغة الثانية :

```
READ m, List
m FORMAT(.....)
```

وتستخدم هذه الصيغة عندما يكون هناك جهاز واحد فقط لادخال المعطيات فلا يكون هناك حاجة للاشارة الى رقم جهاز الادخال n

```
READ 30, Y, X
30 FORMAT (F6.2, F4.1)
```

2.2 عمل الاخراج OUTPUT STATEMENTS

وتستخدم هذه الجملة لطباعة نتائج العمليات الحسابية وقيم المعطيات لغرض التأكد من صحة عملية ادخالها .

a. جملة الطباعة غير المصانة وصيغتها العامة

```
PRINT*, List of variables
PRINT*, A, B, C
```

وتظهر هذه القيم ، عند طباعتها على الصورة التالية :

12.6 3.52 13.2

ويمكن استخدام هذه الصيغة لطباعة العناوين والاسماء :

```
PRINT*, 'A=', A, 'B=', B
```

وتظهر النتائج على الصورة التالية :

A=12.6 B=3.52

b - جملة الطباعة المعيارية:

الصيغة الاولى:

```
WRITE (n,m) List of Variables
m FORMAT (----)
```

حيث

n: رقم جهاز الافراج المستخدم لطباعة النتائج

m: رقم جملة الصيغة التي تبين كيفية طباعة النتائج بالوسط الخارجي.

الصيغة الثانية:

```
PRINT m, List of Variables
m FORMAT (-----)
```

يعني أن هناك جهازاً "معيناً" للافراج، فلا يلزم أن نشير الى رقمة n.

ومثله ذلك:

```
WRITE (6, 30) X, Y
30 FORMAT (2F6.3)
```

وباستخدام الصيغة الثانية

```
PRINT 30, X, Y
30 FORMAT (2F6.3)
```

2.3 جملة الصيغة FORMAT STATEMENTS

وتستعمل لبيان كيفية قراءة أو طباعة المعطيات أو النتائج من قبل جهاز الادخال أو الافراج. وتنقسم جملة الصيغة الى:

a - صيغة I I-FORMAT

وتستعمل هذه الصيغة للثوابت العددية الصحيحة وطريقتها:

nIw

حيث أن

n: عدد مرات التكرار

I: صيغة الثوابت العددية الصحيحة

W : عدد الاعمدة بضمنها الاشارة.

مثال: أقرأ المتغيرات $M=-205$ ، $K=13$

```
READ (5, 30) K, M
30 FORMAT (I3, I5)
```

عند التنفيذ فأن ادخال البيانات يكون:

على النحو الآتي

1	2	3	4	5	6	7	8	9	0	1
1	3		-	2	0	5				

يجب أن يكتب العدد الصحيح في أقصى مكان للحجر وتترك الفراغات

اما اقصى اليسار أو تعتبر الفراغات اصفار، أما في حالة أدخال المعطيات من أقصى اليسار فأن الحاسبة ستقرأ $M=-2050$ ، $K=130$.

1	2	3	4	5	6	7	8	9	0	1
1	3	0	-	2	0	5	0			

مثال:

```
READ (5, 40) I, J
40 FORMAT (I4, I4)
```

or

```
40 FORMAT (2 I4)
```

b- صيغة F F-FORMAT

تستعمل هذه الصيغة للثوابت العددية الحقيقية وكما يلي:

m = n.d

حيث تمثل:

n: عدد رقات الحزر

F: صيغة الثوابت العددية الحقيقية

n : عدد الاعمده التي يشغلها الرقم بضمنها الفاصلة العشرية والاشارة

d : عدد الاعمده التي تقع الى يمين الفاصلة العشرية

مثال : $X=123.5$ ، $Y=3.12$

```
READ (5, 20) X, Y
20 FORMAT (F5.1 , F4.2)
```

```
1 2 3 4 5 6 7 8 9 1
| 1 2 3 . 5 | 3 . 1 2 |
```

c - صيغة X

X-FORMAT

وتستعمل هذه الصيغة لترك الفراغات بين الحقول، وتأخذ الصيغه التاليه:

n X

حيث:

n : عدد الاعمده المطلوب أن تكون فارغة.

مثال:

```
READ (5, 40) A, B
40 FORMAT (F7.3, 2X, F5.1)
```

على فرض أن قيم $A=92.5$ ، $B=-26.0$

فأن بطاقة المعطيات تأخذ الصورة التالية:

```
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
|   9 2 . 5 |  |-2 6 . 0|
```

d - صيغة الحرفيات Literal FORMAT

تستعمل هذه الصيغة في كتابة العبارات او العناوين المطلوب طبعها وتكون عادة بين قوسين على الجانبين

ومثال ذلك

```
WRITE(6,50) A,B                    B=30.5 ، A=20.12
50 FORMAT(2X,' A=',F5.2,3X,'B=',F4.1)
```

وتظهر النتائج على الصورة التالية

```
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9
  A = 2 0 . 1 2     B = 3 0 . 5
```

e. صيغة السطور المتعددة Multiple-line FORMAT

تستعمل هذه الصيغة عند إدخال أو إخراج عدد من المعطيات أو النتائج في أكثر من سطر واحد.

```
WRITE(6,25) A , B
25 FORMAT(1X,'A=',F5.2/1X,'B=',F4.1)
```

```
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9
A = 4 0 . 3 2
B = 2 1 . 2
```

2.4 جمل التعليق والإيضاح Comment Statements

تستخدم هذه الجمل لغرض توضيح بعض اجزاء البرنامج مما يجعل خطوات سير البرنامج واضحة وبيّنة للمبرمج نفسه ولأي شخص آخر يطلع على البرنامج بقصد الاستفادة منه أو لإجراء أي تعديل عليه وتأتي مثل هذه الجمل في أي موقع من البرنامج، وسنستخدم لهذا الغرض الحرف (C) في العمود الاول في بداية جملة التعليق.

مثال:

```
C THIS PROGRAM IS TO COMPUTE ROOTS
C OF A QUADRATIC EQUATION
```

2.5 جملة النهاية END Statement

أن كل برنامج من برامج فورتران يجب أن ينتهي بعبارة النهاية END ولا يجوز اطلاقا كتابة أي عبارة أخرى بعد هذه العبارة. وهي عبارة غير تنفيذية تستخدم لانهاء البرنامج.

مثال: أذا علمت أن قيم كل من X و Y هي 2.65 , 3.16- على الترتيب، اكتب برنامجا تحسب قيمة الاقترانات الآتية:

$$W=\sqrt{2X}$$

$$Z=\sqrt{3X^3-Y^2}$$

$$T=W \sin Z$$

$$U=\frac{T}{W.Z}$$

الحل: يكون البرنامج المطلوب على الصورة التالية:

```
READ X,Y,Z,W,T,U
READ *, X, Y
W=SQRT (2.0 *X)
Z=SQRT (3.0 *X**2 - Y**2)
T=W* SIN(Z)
U=T/(W*Z)
PRINT *, W
PRINT *, Z
PRINT *, T
PRINT *, U
END
```

ويكون نتائج البرنامج على الشكل الآتي

```
W      2.30
Z      3.33
T     -0.45
U     -0.06
```

مثال: اكتب برنامجاً بلغة فورتران لحساب:

1- المتوسط الحسابي A من المعادلة:

$$A=\frac{z_1+z_2+z_3+z_4}{4}$$

2- المتوسط الهندسي G من المعادلة:

$$G=\sqrt[4]{z_1 \cdot z_2 z_3 \cdot z_4}$$

الحل:

```
REAL Z1, Z2, Z3, Z4, A, G
READ*, Z1, Z2, Z3, Z4
A=(Z1+Z2+Z3+Z4)/4.0
G=(Z1*Z2*Z3*Z4)**(1.0/4.0)
PRINT*, A, G
END
```

الفصل الثالث
جمل التحكم والسيطرة

3.1 جملة الانتقال غير الشرطية Unconditional GoTo statement

صيغتها العامة:

GoTo n

حيث (n) رقم الجملة، التي سينتقل الحاسوب لتنفيذها حسب الايعاز GoTo.

3.2 جملة الانتقال الشرطية Conditional GoTo Statement

صيغتها العامة:

GoTo $(n_1, n_2, n_3, \ldots, n_m)$ i

حيث $(n_1, n_2, \ldots, n_m)$ تمثل أرقام جمل يتم الانتقال اليها لتنفيذها حسب الايعاز GoTo أما (i) فتمثل متغيراً "عددياً" صحيحاً، وتكون دائماً موجبة، بحيث:

اذا كانت قيمة (i=1) فان الجملة التالية في التنفيذ هي جملة رقم (n_1)

أما اذا كانت قيمة (i=2) فان الجملة التالية في التنفيذ هي جملة رقم (n_2) وهكذا.

مثال/ اكتب برنامجاً "لحساب قيمة (W) من المعادلات الآتية إذا علمت أن قيم كل من X و Y و Z، معلومة لديك

K=1	W = 10 Cos(XYZ)
K=2	W = X + Y + Z
K=3	W = X + 2Y − 4Z + 5
K=4	W = 3XY − e^{Y} − 2

```
      INTEGER K
      REAL X, Y, Z, W
      READ *, K
      READ *, X, Y, Z
      GoTo (1, 2, 3, 4) K
1     W = 10.0 * COS(X * Y * Z)
      GoTo 10
2     W = X + Y + Z
      GoTo 10
3     W = X + 2.0 * Y - 4.0 * Z + 3.0
      GoTo 10
4     W = 3.0 * X * Y + EXP(Y) + 2.0
10    PRINT *, 'W=', W
      END
```

3-3 جملة اذا الحسابية Arithmetic IF Statement

صيغتها العامة:

IF (X) n_1, n_2, n_3

حيث (X) تمثل تعبيراً رياضياً قد تكون قيمته سالبة أو صفراً أو موجبة

ففي حالة (X) سالبة تكون جملة التنفيذ التالية هي الجملة رقم (n_1)

وفي حالة (X) صفراً تكون جملة التنفيذ التالية هي الجملة رقم (n_2)

وفي حالة (X) موجبة تكون جملة التنفيذ هي الجملة رقم (n_3).

مثال / اكتب برنامجاً لحساب قيمة Y وكما يلي:

$$Y = \begin{cases} 3.5 + X^2 + \dots + \sqrt{X} & X = 2 \\ 3.5 + \sqrt{X} & \\ 3.0 + X^4 & X > 2 \end{cases}$$

```
      REAL X,Y
      READ*, X
      IF (X-2.0) 10,20,30
10    Y= 3.5 + X**2 + 9.0 *X **3 + SQRT(X)
      GoTo 50
20    Y=3.5+SQRT(X)
      GoTo 50
30    Y=3.0+ X**4
50    PRINT*, X,.Y
      END
```

3.4 جملة اذا المنطقية Logical IF statement

وصيغتها العامة:

IF (a) S

حيث (a) تعبير منطقي قيمته (TRUE) أو (FALSE)

(S) جملة تنفيذية أو رقم الجملة التالية في التنفيذ، اذا كانت قيمة التعبير المنطقي (TRUE) وعندما تكون قيمة التعبير المنطقي (FALSE) لا يتم تنفيذ (S)، ويستمر البرنامج.

والتعبير المنطقي يتكون من مجموعة من الثوابت والمتغيرات العددية ونستخدم له رموز العمليات المنطقية الاتية:

.EQ.	يساوي
.NE.	لا يساوي
.GT.	اكبر من
.LT.	أصغر من
.GE.	اكبر من أو يساوي
.LE.	أصغر من أو يساوي

مثال/ اكتب برنامجاً لحساب قيمة المقدار التالي:

$$W = \begin{cases} 1 + Z^2 & Z < 0 \\ 1 + Z & Z \geq 0 \end{cases}$$

```
      REAL  Z, W
      READ *, Z
      IF ( Z .GE. 0.0 ) GoTo 20
      W = 1.0 + Z**2
      GoTo 40
20    W = 1.0 + Z
40    PRINT*, Z, W
      END
```

IF) بالإضافة إلى ذلك يمكن أن نجمع أكثر من عملية منطقية في جملة الواحدة، ويسمى التعبير المنطقي في هذه الحالة تعبيراً "منطقياً" مركباً" ونستخدم لذلك أدوات الربط (.AND.)، (.OR.).

مثال/

```
Y = 1 + 2X     (2 < X < 10)
IF ( X .GT. 2.0 .AND. X .LT. 10.0 )  Y = 1.0 + 2.0 * X
```

3.5 جملة اذا... فان

IF... THEN statement

```
IF ( Condition ) THEN
   ---
   ---   } YES
   ---
END IF
```

حيث (condition) هو شرط او اكثر مطلوب تحقيقه . فاذا تحقق الشرط فانه يتم تنفيذ جميع الجمل التالية لجملة (IF) ولغاية جملة (ENDIF) واذا لم يتحقق الشرط فانه يتم الانتقال الى الجملة التالية لجملة (ENDIF)

مثال / اكتب برنامجاً لحساب (W) من المعادلات الآتية :

$$W = \begin{cases} AX^2+5 & A \leq 0 \\ 2 & A > 0 \end{cases}$$

```
REAL A, X, W
READ*, A, X
IF( A .LE. 0.0) THEN
W=A**X**2+5-0
GoTo 10
END IF
W=2.0
10PRINT*, W
END
```

IF...THEN...ELSE Structure

3-6 بنية إذا... فأن... والا صيغتها العامة :

```
IF ( condition) THEN
---
---  } Yes
ELSE
---
---  } No
END I
```

حيث (condition) هو شرط أو اكثر مطلوب تحقيقه . فاذا تحقق الشرط فانه يتم تنفيذ جميع الجمل التالية لجملة (IF) ولغاية جملة (ELSE) . واذا لم يتحقق الشرط فانه يتم تنفيذ جميع الجمل التالية لجملة (ELSE) ولغاية جملة (ENDIF) .

مثال/ اكتب برنامجاً لحساب (w) من المعادلات الآتية :

$$w = \begin{cases} AX^2+5 & A \leq 0 \\ 2 & A > 0 \end{cases}$$

```
REAL A, X, W
READ*, A, X
IF(A.LE. 0.0) THEN
W= A* X**2+5.0
GoTo 10
ELSE
W= 2.0
END IF
10 PRINT*, A, W
END
```

3-7 بنية اذا المتداخلة NESTED IF structure وصيغتها العامة :

```
IF ( condition1) THEN
--- { Yes 1
---
ELSEIF ( Condition 2) THEN
--- { Yes 2
ELSE IF ( Condition 3) THEN
--- { Yes 3
```

```
ELSEIF (condition n) THEN
... } Yes n
ELSE
... } NO
ENDIF
```

مثال / اكتب برنامجاً لحساب عدد رينولدز (RE) لسائل تجري بسرعة (V) ولزوجة حركية (NU) داخل أنبوب قطره (D)، ومن ثم بين نوع الجريان حسب التصنيف التالي:

LAMINAR	$RE \leq 2100$
TRANSITION	$2100 < RE < 4000$
TURBULENT	$RE \geq 4000$

علماً أن $RE = VD/NU$.

```
CHARACTER* 10 TYPE
REAL V, NU, D, RE
READ*, V, NU, D
RE=(D*V)/NU
IF (RE .LE. 2100.0) THEN
TYPE = 'LAMINAR'
ELSEIF (RE.LT. 4000.0 .OR. GT. 2100.0) THEN
TYPE = 'TRANSTION'
ELSE
TYPE= 'TURBULENT'
END IF
PRINT*, TYPE
END
```

الفصل الرابع
عمل الدوران والتكرار

4.1 جملة الدوران والتكرار Do statement

صيغتها العامة:

```
Do n, INDEX = m1, m2, m3
...
...
...
n CONTINUE
```

حيث:

(INDEX) يمثل دليل عدد مرات التكرار لحلقة الدوران Do.

(m_1) يمثل القيمة الابتدائية للعداد INDEX.

(m_2) يمثل القيمة النهائية للعداد INDEX.

(m_3) يمثل قيمة الزيادة المنتظرة في قراءة العداد INDEX في كل دورة.

(n) يمثل رقم آخر جملة في حلقة الدوران Do، وهي CONTINUE.

مثال / اكتب برنامجاً لايجاد قيمة (w) في التعبير الحسابي:

$$w = 1 + z + 4z^3$$

إذا علمت أن قيم (z) هي على التوالي: 0.1، 0.2، 0.3 ... 0.9، 1.0

```
Do 20, K = 1, 10
Z = K / 10.0
W = 1.0 + Z + 4.0 * Z**3
PRINT W, Z
20 CONTINUE
STOP
END
```

4.2 قواعد استخدام جملة Do

1- يمكن هنا الفارزة بعد الرقم (n) في جملة Do .

2- اذا لم تذكر قيمة الزيادة في العداد (m_3) ، ففي هذه الحالة سوف تأخذ (1) وبصورة أوتوماتيكية.

3- يمكن ان تكون قيمة ($m_1=0$).

4- اذا كانت قيمة ($m_1>m_2$) فينبغي ان تكون قيمة الزيادة (X) سالبة.

5- لايمكن ان تكون ($m_3=0$) .

6- يمكن انهاء حلقة الدوران بالجملة التي تسبق CONTINUE .

7- يجوز الانتقال من داخل حلقة الدوران الى خارجها ، باستخدام جملة انتقال قبل استكمال مرات الدوران ، اذا لزم الامر.

```
    Do 10 I=1,10
    READ*, Y
    X=FLOAT(I)
    IF (SQRT(Y).GT.SQRT(X)) GoTo 20
    PRINT*, Y
10  CONTINUE
20  STOP
    END
```

8- لايجوز الانتقال من خارج حلقة الدوران الى داخلها .

9- لايمكن استخدام اي جملة في مدى حلقة الدوران لتغيير قيمة المتغير (INDEX) ، كما في المثال التالي!

```
    Do 20, I=1,100
    X=FLOAT(I)
    IF (I.GT.50)  I=I+4
20  CONTINUE
```

10- لايمكن ان تتفرع حلقة الدوران بأي من الجمل الآتية:
a- جملة التوقف النهائي أو اللحظي (STOP ، PAUSE).
b- جمل التحكم والسيطرة (IF ، GoTo).
c- جملة العودة من البرنامج الفرعي الى البرنامج الرئيس (RETURN)
d- جملة (Do).

ملاحظة/ قد نحتاج الى استعمال بعض الجمل مثل (IF ، GoTo ، STOP) لتكون آخر جملة في حلقة الدوران، الا ان ذلك يتعارض مع القاعدة (10) ولتجنب مثل هذا التعارض يمكن ان نستعمل جملة الاستمرار CONTINUE قبل جملة (IF) وكما في المثال التالي

```
    Do 20, J=1, 100
    READ*, A
    IF(A.GT.5.0)  Y = A**3 + 10.0
    IF(A.LE.0.0)  STOP
20  CONTINUE
```

3-4 حلقات الدوران المتداخلة Nested Do loops

صيغتها العامة:

```
Do 10, I=1, 50
  Do 20, J=1, 1000
    Do 30, K=1, 10
30  CONTINUE
    ---
20  CONTINUE
    ---
10  CONTINUE
```

وتسمى أصغر حلقات الدوران بحلقات الدوران الداخلية Inner Do loops
في حين تسمى الحلقات الأخرى بحلقات الدوران الخارجية Outer Do loops.
ومن الممكن، لأكثر من حلقة دوران، أن تنتهي بالجملة نفسها وكما يلي:

```
Do 10, I = 1,5
Do 10, J = 1, 10
Do 10, K = 1, 5
- - - -
- - - -
- - - -
10 PRINT*, I, J, K
```

مثال/ اكتب برنامجاً بلغة فورتران لقراءة العدد الموجب (N)، ثم حساب المجموع (SUM) حسب العلاقة:

$$SUM = \sum_{i=1}^{N} \frac{1}{i}$$

```
READ*, N
SUM = 0.0
Do 10, I = 1, N
10 SUM = SUM + 1.0 / FLOAT(I)
PRINT*, N, SUM
STOP
END
```

مثال/ اكتب برنامجاً لحساب قيمة الاقتران:

$$Z = F(X,Y) = X^2 + XY - Y^2$$

```
      REAL X, Y, Z
      DO 40, I = 1, 13
      X = -3.0 + 0.5 * FLOAT(I-1)
      DO 40, J = 1, 13
      Y = -3.0 + 0.5 * FLOAT(J-1)
      Z = X * X + X * Y + Y * Y
40    PRINT *, X, Y, Z
      STOP
      END
```

الفصل الخامس
المتغيرات المؤشرة (المصفوفات)

5.1 جملة الابعاد DIMENSION statement

عند استعمال المتغيرات المؤشرة يجب استعمال جملة الابعاد لحجز العدد المطلوب من الخازن في وحدة الذاكرة الرئيسية، لخزن قيم العناصر فيها، وتعتبر هذه الجملة غير تنفيذية.

a- المتغيرات المؤشرة ذات البعد الواحد

صيغتها العامة:

DIMENSION name1(size1), ..., name n (size n).

حيث تمثل (name1, ..., nameN) اسماء المتغيرات المؤشرة بينما تمثل (size1, size2, ..., sizeN) عدد المواقع المطلوب حجزها في وحدة الذاكرة.

مثال/

DIMENSION X(10), Y(20)

b- المتغيرات المؤشرة متعددة الابعاد

تأخذ جملة الابعاد لمتغير مؤشر ذي بعدين الصورة العامة التالية:

DIMENSION name1(n1,m1), name2(n2,m2)...

حيث يمثل (n1) عدد الصفوف للمتغير المؤشر (name1) بينما يمثل (m1) عدد الاعمدة للمتغير المؤشر (name1).

مثال/

DIMENSION A(2,3), B(3,2)

تعني ان عدد صفوف المصفوفة (A) هو (2) وعدد الاعمدة هو (3)، اما بالنسبة للمصفوفة (B)، فان عدد الصفوف هو (3) بينما عدد الاعمدة هو (2).

مثال/

DIMENSION X(...), ABC(...3), V(10,5...)

5.2 جمل الادخال والاخراج للمتغيرات المؤشرة

a- المتغيرات المؤشرة ذات البعد الواحد

الصيغة العامة لجملتي القراءة والطباعة هي:

عند القراءة:

READ*, (Array name(INDEX), INDEX=m_1, m_2, m_3)

وعند الطباعة:

PRINT*, (Array name(INDEX), INDEX=m_1, m_2, m_3)

حيث:

(Array name) هو اسم المتغير المؤشر المشار اليه في جملة DIMENSION

(INDEX) مؤشر المتغير المؤشر.

(m_1) القيمة الابتدائية.

(m_2) القيمة النهائية.

(m_3) الزيادة في قيمة المؤشر.

مثال/

```
DIMENSION X(10)
READ*, (X(I), I=1,10,2)
```

كما يمكن استخدام حلقات التكرار المتداخلة وكما يلي:

```
DIMENSION X(10)
DO 10, I=1, 10, 2
READ*, X(I)
10 CONTINUE
```

b- المتغيرات المؤشرة متعددة الابعاد

لقراءة أو طباعة مجموعة عناصر بمقدوفة ذات بعدين أو اكثر يمكننا ان نستخدم صيغة التكرار الضمنية وكما يلي:

عند القراءة: READ*, ((Name(I,J), J=1, m), I=1, n)

وعند الطباعة: PRINT*, ((Name(I,J), J=1, m), I=1, n)

مثال /

```
DIMENSION A(10,10)
READ*, ((A(I,J), J=1, M), I=1, N)
```

ان القراءة تكون صفاً صفاً

كما يمكن استخدام حلقات التكرار المتداخلة لقراءة وطباعة المصفوفات وكما يلي :

```
DIMENSION A(10,10)
Do 2, I=1, N
Do 2, J=1, M
READ*, A(I,J)
2 CONTINUE
```

مثال / اكتب برنامجاً لحساب متوسط أعمار AGES أو AVERAGE صف عدد طلابه (30) طالباً.

```
DIMENSION A(30)
SUM=0.0
Do 20, I=1, 30
READ*, A(I)
20 SUM=SUM+A(I)
AVERAGE = SUM/30.0
PRINT*, AVERAGE
END
```

مثال / اكتب برنامجاً لقراءة عناصر المصفوفة A(N,M) ثم طباعة حاصل ضربها في ثابت عددي (K).

```
DIMENSION A(10,10), B(10,10)
READ*, K
Do 15, I=1, 10
```

```
      Do 15, J=1, 10
      READ*, A(I,J)
      B(I,J) = K* A(I,J)
15    CONTINUE
      Do 25, I=1, 10
      Do 25, J=1, 10
      PRINT*, B(I,J)
25    CONTINUE
      END
```

مثال/ جد القيمة التقريبية للتكامل $\int_0^{\pi} \sin x \, dx$ باستخدام العلاقة التالية:

$$SUM = \int_0^{\pi} f(x)\,dx = \sum_{k=1}^{N} f(X_k)\,\Delta X$$

الحل/

$$\Delta X = \frac{\pi - 0}{N} = \frac{3.14}{N}$$

$y = \sin X$

```
      DIMENSION Y(100)
      DX = 3.14/100.0
      SUM = 0.0
      Do 5, I=1, 100
      Y(I) = SIN( FLOAT(I) *DX) * DX
      SUM = SUM + Y(I)
5     CONTINUE
      PRINT*, SUM
      END
```

مثال/ اكتب برنامجاً لحساب المصفوفة C الناتجة من ضرب المصفوفتين A(3x2) في B(2x3).

الحل/ من قوانين المصفوفات نجد ان المصفوفة C الناتجة ذات بعدين هما (3X3).

```
      DIMENSION A(3,2), B(2,3), C(3,3)
      READ*, M, K, N
C     READ MATRIX A
      READ*, ((A(I,J), J=1,K), I=1,M)
C     READ MATRIX B
      READ*, ((B(I,J), J=1,N), I=1,K)
      DO 4, I=1,M
      DO 4, J=1,N
      C(I,J)=0.0
      DO 5, L=1,K
5     C(I,J)=C(I,J)+A(I,L)*B(L,J)
4     CONTINUE
      PRINT*, ((C(I,J), J=1,N), I=1,M)
      END
```

الفصل السادس
البرامج الفرعية

6.1 مقدمة

تتألف البرامج في العادة من جزء رئيسي يعرف بالجزء الرئيسي (Main Program) واجزاء فرعية تعرف (Sub-program) . أن الاجزاء الفرعية هذه هي برامج متكاملة قائمة بذاتها لانجاز عمليات معينة تضاف الى الجزء الرئيسي من البرنامج لانجاز عمليات قد تكرر الكثير من مرة في البرنامج . وهي بذلك تزيد من كفاءة البرنامج لما توفره من اماكن خزن في ذاكرة الحاسبة التي يتطلب استخدامها في حالة تكرار العمليات المتشابهة .

6.2 انواع البرامج الفرعية

هناك أربعة انواع رئيسية من البرامج الفرعية هي !

a - الاقترانات المكتبية Library Function

فقد وردت تفصيل سابق في الفقرة (1.6) الدوال الحسابية .

b - الاقترانات الداخلية Internal Functions

وسميت بالداخلية لانه يتم تعريفها داخل البرنامج الرئيس وتستعمل مثل هذة الاقترانات عندما لاتكون مثل هذة الاقترانات موجودة في مخزن الاقترانات المكتبية ، حيث يمكننا تكرار استعمالها داخل البرنامج بعد تعريفها مرة واحدة في بداية ، وذلك بالاشارة الى اسمها فقط في داخل البرنامج وعرفتها العادة !

NAME (a1 ... an) = Arithmetical Expression

حيث NAME اسم الدالة و (a1 ... an) اسماء المتغيرات الدالة . أما (Arithmetical Expression) فتمثل التعبير الرياضي المراد استخدامه في عمل الدالة .

مثال/

```
READ *, X, Y, V, W, Q
PSQ(X,Y) = SQRT(X**2 + Y**2)
R1 = PSQ(V,W) * 0.5 + V - 1.0
R2 = (Q*W) / PSQ(X*0.1, -1.0)
PRINT *, R1, R2
END
```

الاقترانات الخارجية

وسميت هذه البرامج بالخارجية لأنها تكون خارج البرنامج الرئيس وهي تعتبر برامج متكاملة ومستقلة بذاتها وتستعمل مثل هذه الاقترانات عند الحاجة لحساب نتيجة واحدة من مجموعة عمليات حسابية وصيغتها العامة:

```
FUNCTION name (a1, a2, ----, an)
. . .
. . .
. . .
RETURN
END
```

وتنتهي بجملة العودة (RETURN) حيث تعود بعدها للبرنامج الرئيس.

مثال/ الاقتران G(X,Y,n) معرّف على النحو التالي:

$$G = \begin{cases} X^n \cdot Y^n, & n \neq 0 \\ 1, & n = 0 \end{cases}$$

اكتب (z) بالعبارة التالية

$$z = G^2(r,s,t) + 2$$

```
READ*, R, S, T
Z = G (R, S, T) **2 + 2.0
PRINT*, Z
END
FUNCTION G(X, Y, N)
IF(N.EQ.0) GoTo 2
1 G = X*Y **N
RETURN
2 G = 0.0
RETURN
END
```

SUBROUTINES البرامج الروتينية الفرعية

وتستعمل عند الحاجة لحساب اكثر من نتيجة واحدة في البرنامج الرئيسي كل نتيجة من هذه النتائج تحتاج الى مجموعة من العمليات الحسابية المتكررة لذا نضعها في برامج فرعية مستقلة عن البرنامج الرئيس وصيغتها العامة:

```
SUBROUTINE name (X1, X2, --- Xn, Y1, Y2, --- Yn)
- - -
- - -
- - -
RETURN
END
```

حيث يجب ان يبدأ البرنامج الروتيني الفرعي بكلمة (SUBROUTINE) يتبعها اسم البرنامج الروتيني الفرعي (name) واما المتغيرات (X_1 X_2 --- X_n) فتمثل المدخلات في البرنامج الرئيسي واما (Y_1 Y_2 --- Y_n) فتمثل المخرجات.

وسيتم استدعاء البرنامج الروتيني الفرعي من البرنامج الرئيسي بواسطة جملة الاستدعاء التي تأخذ الصورة العامة التالية :

CALL name ($V_1, V_2, \ldots, V_n, Z_1, Z_2, \ldots Z_n$)

حيث يمثل (name) هو اسم البرنامج الروتيني الفرعي نفسه الوارد بالعبارة ($V_n \ldots V_2, V_1$) هي متغيرات عددية تقابل المدخلات ($X_n \ldots X_2, X_1$) في البرنامج الفرعي بينما تمثل ($Z_n \ldots Z_2, Z_1$) متغيرات عددية تقابل المخرجات ($Y_n \ldots Y_2, Y_1$) في البرنامج الفرعي.

مثال / باستخدام البرنامج الروتيني الفرعي ، اجمع (N) من الاوزان ثم جد المعدل

```
DIMENSION WEIGHT(30)
DO 20, I=1,30
20 READ*, WEIGHT(I)
CALL AV(30,WEIGHT,AVERGE)
PRINT*, AVERGE
STOP
END
SUBROUTINE AV(N,Y,Z)
DIMENSION Y(N)
SUM=0.0
DO 1, I=1,N
1 SUM=SUM+Y(N)
Z=SUM/FLOAT(N)
RETURN
END
```

6.3 عبارة الخزن المشترك COMMON Statement

ان عبارة الخزن المشترك تمكن البرامج الجزئية المختلفة من استخدام نفس المعلومات المخزونة في جزء مشترك من الذاكرة.

ويقسم الى قسمين هما!

a - الخزن المشترك غير المعنون

والصيغة العامة لهذا النوع هي!

COMMON List

حيث تمثل (List) اسماء المتغيرات المشتركة.

مثال/

DIMENSION A(3,100) , Q(10,10)

COMMON B, A, PN, Q

ويمكن كتابتها بعبارة واحدة وكالاتي!

COMMON B, A(3,100), PN, Q(10,10)

b - الخزن المشترك المعنون

ان الفرق بينه وبين النوع الاول (غير المعنون)، أنه يسجل ويخصص عنوانا لكل مخزن من مخازن (COMMON) حيث يوضع اسم هذا العنوان بين خطين مائلين.

الصيغة العامة!

COMMON /Label/ List

حيث ان (label) تمثل اسم جزء الخزن المشترك الذي يحتوي على المتغيرات المذكورة في القائمة المصاحبة له (List)

مثال/

COMMON /block/ A, B, C

مثال/ استخدم البرنامج الرئيسي الفرعي وجملة التخزين المشترك
حيث مجموع (N) من الاوزان ثم ايجاد المعدل.

```
      DIMENSION WEIGHT (30)
      COMMON K, WEIGHT, AVERGE
      DO 20, I=1, 30
20    READ*, WEIGHT(I)
      CALL AV (30, WEIGHT, AVERGE)
      PRINT*, AVERGE
      STOP
      END
      SUBROUTINE AV(N, Y, Z)
      COMMON N, Y, Z
      DIMENSION Y(30)
      SUM=0.0
      DO 10, I=1, 30
10    SUM=SUM+Y(I)
      Z=SUM/FLOAT(N)
      RETURN
      END
```

مثال/ اكتب برنامجاً بلغة فورتران لحساب قيمة A من المعادلة التالية:

$A = X / D(X, Y, Z)$

ثم استخدم برنامج اقتران جزئي في حساب قيمة (D) والمعرفة على الشكل التالي:

$D(X, Y, Z) = \sqrt{X^2 + Y^2 + Z^2}$

الحل/

```
READ*, X, Y, Z
A = X / D(X, Y, Z)
PRINT*, A
STOP
END
FUNCTION D(R,S,T)
D(R,S,T) = SQRT(R*R+S*S+T*T)
RETURN
END
```

مثال/ اذا كانت A مصفوفة ذات بعد واحد وعدد عناصرها (N)، اكتب برنامجاً بلغة فورتران لايجاد مجموع هذة العناصر. استخدم برنامج الروتين الفرعي في الحل.

الحل/

```
DIMENSION A(10)
DO 10, I = 1, 10
READ*, A(I)
CONTINUE
CALL SUB(10, A, SUM)
PRINT*, SUM
STOP
END
```

```
SUBROTINE  SUB( N, B, Z)
DIMENSION  B(N)
Z=0.0
Do 20, I = 1, N
20 Z= Z+ B(I)
RETURN
END
```

مثال / اكتب برنامجاً بلغة فورتران لحساب z من المعادلة التالية:

$$Z = Y^2 + Y + 1$$

ثم استخدم برنامجاً فرعياً لحساب Y ، والمعرفة على النحو الآتي:

$$Y = \begin{cases} 1 + X^2 & X < 0 \\ 0 & X = 0 \\ 1 - X^2 & X > 0 \end{cases}$$

الحل/

```
READ*, X
CALL  AV (X, Y)
Z= Y*Y + Y + 1.0
PRINT*, Z
STOP
END
SUBROTINE  AV( A, B)
IF (A.GT. 0.0) THEN
B = 1.0 + A*A
ELSEIF (A.LT. 0.0) THEN
B = 1.0 - A*A
ELSE
```

```
B=0.0
END IF
RETURN
END
```

مثال/ اكتب برنامجاً بلغة فورتران لحساب مجموع (N) من الحدود والتالي

$$\frac{1}{W} + \frac{1}{W^2} + \frac{1}{W^3} + \cdots + \frac{1}{W^N}$$

ثم استخدم برنامجاً فرعياً لحساب W من المعادلة التالية:

$$W(X,Y) = X^2 + Y^2 + XY$$

الحل/

```
READ*, X, Y, N
CALL FUN (X, Y, W)
SUM=0.0
DO 10, I=1, N
SUM=SUM+1.0/W**I
PRINT*, SUM
STOP
END
SUBROUTINE FUN (A, B, Z)
Z=A*A+B*B+A*B
RETURN
END
```

الفصل السابع
تطبيقات هندسية

7.1 ايجاد جذور المعادلات

هنالك طرق مختلفة لايجاد جذور معادلة ما، ومن هذه الطرق هي:

a. طريقة المحاولات المتكررة Iterative Method

لايجاد جذور المعادلة $f(X)=0$، بطريقة المحاولة والخطأ، يمكن تلخيص خطوات خوارزمية الحل، بما يأتي:

1- اعد ترتيب المعادلة لتصبح على النحو: $X_A=G(X_B)$

2- اقرأ قيمة أولية لـ XB وحدد الدقة ϵ.

3- ضع I=1

4- احسب قيمة X_A وفقاً للمعادلة $X_A=G(X_B)$، ثم اضبع قيمتي XA، XB

5- اذا كانت: $|X_B-X_A|\leq\epsilon$، فاذهب الى الخطوة 7 والا فاذهب الى الخطوة 6.

6- أبدل XA بـ XB وأضف (1) الى قيمة (I) ثم عد الى الخطوة 4.

7- اطبع جذر المعادلة XA.

8- توقف.

مثال/ اكتب برنامجاً بلغة فورتران لايجاد جذر المعادلة:

$$f(X)=5-25X-\sec\sqrt{0.68X}=0$$

الحل/ يمكن اعادة ترتيب المعادلة لتصبح على النحو التالي:

$$X_A=(\sec\sqrt{0.68X_B})/5-25$$

```
C  ITERATIVE SOLUTION OF F(X)=0
   REAL *8 I
   ERROR = 0.0001
   G(X) = (1.0/COS(SQRT(0.68*X)))/5-25
```

```
I=1
4 XA=G(XB)
PRINT*, I, XB, XA
5 IF (ABS (XA-XB).LE. ERROR) GoTo 7
6 XB=XA
I=I+1
GoTo 4
7 PRINT*, 'ROOT IS', XA
PRINT*, 'ITERATION NO. IS', I
STOP
END
```

b - طريقة نيوتن رافسون NEWTON-RAPHSON METHAD

تقوم فكرة هذه الطريقة على إيجاد جذور المعادلة $f(x)=0$ ، على القاعدة التالية

$$X_{i+1} = X_i - \frac{f(X_i)}{f'(X_i)}$$

حيث:

X_i : تمثل قيمة أولية مفترضة لـ X .

X_{i+1} : تمثل قيمة لـ X أقرب من X_i للجذر.

$f(X_i)$: تمثل قيمة دالة $f(x)$ عند X_i.

$f'(X_i)$: تمثل مشتقة دالة $f(x)$ عند X_i .

تتكون خوارزمية نيوتن رافسون من الخطوات الآتية :

- ابدأ بقيمة أولية لقيمة X_i وأقل قيمة لهامش الخطأ ε ، وعدد المحاولات

- ضع $i=1$

- احسب قيمة X_{i+1} من القاعدة $X_{i+1} = X_i - \frac{f(X_i)}{f'(X_i)}$

4- اطبع قيمة i ، X_i ، X_{i+1} .

5- اذا كانت $\left|\frac{X_{i+1}-X_i}{X_{i+1}}\right| < \epsilon$ فاذهب إلى الخطوة 7 وإلا فاذهب إلى الخطوة 6.

6- أضف 1 الى قيمة i وضع قيمة X_i الجديدة تساوي قيمة X_{i+1} المحسوبة

7- اذا كان $i < N$ عد إلى الخطوة 3 وإلا فاذهب الى الخطوة 8.

8- اطبع جذر المعادلة X_{i+1}

9- توقف

مثال/ أوجد جذور المعادلة:

$$f(X) = \cos X - X = 0$$

بإستخدام طريقة نيوتن رافسون.

الحل/

$$f'(X) = -\sin X - 1$$

```
C   NEWTON RAPHSON SOLUTION OF F(X)=0
READ*, XI, N
ERROR = 0.0001
F(X) = COS(X) - X
F1(X) = -SIN(X) - 1.0
I = 1
3 XIP1 = XI - F(XI) / F1(XI)
PRINT*, I, XI, XIP1
IF (ABS((XIP1 - XI) / XIP1) .LE. ERROR) GOTO 5
6 I = I + 1
XI = XIP1
IF (I .LT. N) GOTO 3
PRINT*, 'FUNCTION DOES NOT CONVERGE'
```

```
GoTo 9
3 PRINT*, 'ROOT IS', X+D1
PRINT*, 'ITERATION NO IS', I
9 STOP
END
```

س١/ اكتب برنامجاً بلغة فورتران لايجاد جذور المعادلة التالية:

$$2X^2 + 1 - e^X = 0$$

استخدم طريقة المحاولات المتكررة في الحل.

س٢/ اكتب برنامجاً بلغة فورتران لايجاد جذور المعادلة التالية:

$$X^2 + 10 \cos X = 0$$

استخدم طريقة نيوتن رافسون في الحل.

7.2 حل المعادلات الآنية (طريقة جاكوبي)

لحل منظومة من المعادلات الآنية باستخدام طريقة جاكوبي، ينبغي ترتيب هذه المعادلات، بحيث نختار المعادلة التي فيها أكبر معامل لـ X_1 ثم نكتبها بحيث يكون X_1 في الطرف الأيسر من المعادلة الأولى، بدلالة X_2 X_3 ..... X_n وهكذا بالنسبة لـ X_2 و X_3 ..... X_n في المعادلات الثانية والثالثة ..... الخ. وهذه الخطوة تعطينا منظومة المعادلات السابقة على نحو يكون حاصل ضرب عناصر قطر المصفوفة الناتجة المطلق، أكبر ما يمكن.

مثال / حل منظومة المعادلات الآنية الآتية باستخدام طريقة جاكوبي:

$$X_1 + 4X_2 + X_3 = 7$$
$$2X_1 - X_2 - 3X_3 = 15$$
$$X_1 + 6X_2 - X_3 = 13$$

وبعد ترتيب المعادلات ينتج ما يأتي:

المعادلة الأكبر معامل لـ X_1 : $X_1 = (15 + X_2 + 3X_3)/2$

معادلة الأكبر معامل لـ X_2 : $X_2 = (13 - X_1 + X_3)/6$

المعادلة المتبقية لـ X_3 : $X_3 = (7 - X_1 - 4X_2)/1$

وبعد ذلك نفرض قيماً أولية لـ X_1، X_2، X_3 ولتكن 0، 0، 0، ثم نعوض في المعادلات أعلاه، لينتج عندنا قيم جديدة لـ X_1، X_2، X_3 كالتالي

$X_1 = (15 + 0 + 3(0))/2 = 7.5$

$X_2 = (13 - 0 - 0)/6 = 2.167$

$X_3 = (7 - 0 - 4(0))/1 = 7$

تكرر الخطوة السابقة بتعويض القيم الجديدة في المعادلات لنحصل باستخدام قيم جديدة أخرى، وهكذا نستمر الى أن تتقارب جميع القيم

في جميع قيم X_1، X_2، X_3 بين خطوة وخطوة تالية، صفراً "جداً"، واقل من قيمة ضابط الدقة ε، حينئذ تكون قد حصلنا على الحل المطلوب.

تتكون خوارزمية جاكوبي من الخطوات الآتية:

1. رتب المعادلات للمتغيرات، بحيث يكون معامل قطري.
2. ضع قيمة لضابط الدقة.
3. ابدأ بقيمة 1، لعداد محاولات التكرار k أي ضع (K=1)
4. لكل قيمة من قيم i حيث يأخذ i القيم من 1 إلى n (عدد المتغيرات) أحسب:

$$X_i^k = \left[- \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} X_j^{(k-1)} + b_i \right] / a_{ii}$$

حيث: a_{ij}، a_{ii}، b_i معروفة من قبل، k: رقم المحاولة الجديدة و k-1: رقم المحاولة القديمة التي قبلها.

5. اذا كان الفرق لكل قيمة من قيم X_i، بين المحاولة القديمة والمحاولة الجديدة، اقل من ε أي ان: $|X_i^k - X_i^{k-1}| < \varepsilon$ فاذهب الى الخطوة 6 والا فاذهب الى الخطوة 4، واضف 1 الى قيمة k، ثم كرر العملية.
6. اطبع النتائج.
7. توقف.

```
C JACOBI ITERATIVE METHOD
  DIMENSION A(3,4), X1(3), X2(3)
  EPS=1.E-3
  N=100
  READ*,
  M=N+1
  READ*,((A(I,J),J=1,M),I=1,N)
```

```
      READ*, ( X1(I) , I = 1, N)
      PRINT*, ( X1(I), I = 1, N
      K=1
2     ERR=0.0
      Do 10, I=1, N
      Do 11, J = 1, N
      S=0.0
      IF ( I .EQ. J) Go To 11
      S=S-A(I,J) * X1(J)
11    CONTINUE
      S=(S+ A (I, N+1)) /A(I,I)
      ERR=ERR+S*S
10    X2(I) = X1(I) +S
      ERR = SQRT ( ERR)
      PRINT*, K, (X2 (I), I=1, N)
      IF( ERR .LE. EPS) Go To 4
      K=K+1
      IF ( K .GT. NN) Go To 5
      Do 12, I = 1, N
2     X1(I) = X2(I)
      Go To 2
1     PRINT*, 'CONVERGENCE OCCURS AT K= ', K
      STOP
5     PRINT*, 'FAILURE OCCURS IN N= ', NN
      STOP
      END
```

5X1 - X2 + X3 = 10 : أوجد باستخدام طريقة جاكوبي

$$X_1 + X_2 + 5X_3 = -1$$
$$2X_1 + 4X_2 = 12$$

ج/ حل المعادلتين الآنيتين الآتيتين، مستخدماً طريقة جاكوبي التكرارية، وحدد عدد مرات التكرار المطلوبة، لإيجاد الحل:

$$X_1 + 2X_2 = 3$$
$$3X_1 + X_2 = 4$$

7.3 التفاضل العددي Numerical Differentiation

تعتبر طريقة الفرق المركزي، من أشهر الطرق العددية المستعملة، في تمثيل المشتقات التفاضلية الأولى والثانية والثالثة... الخ.

ويعبر عن المشتقة الأولى كما يلي:

$$f'(x) = \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x}$$

أما المشتقة الثانية، فتمثل كالتالي:

$$f''(x) = \frac{f(x+\Delta x) - 2f(x) + f(x-\Delta x)}{(\Delta x)^2}$$

مثال / أكتب برنامجاً لحساب المشتقة التفاضلية الأولى والثانية، للاقتران: $f(x) = x\, e^x$ ، ثم جد قيمتيهما عند $X=1$.

```
C NUMERICAL DERIVATIVE BY CENTRAL DIFFERENCE
  FUNC(X) = X * EXP(X)
  READ*, X, DX
  F1 = (FUNC(X+DX) - FUNC(X-DX))/(2.0*DX)
  F2 = (FUNC(X+DX) - 2.0*FUNC(X) + FUNC(X-DX))/(X*DX)
  PRINT*, X=, X, 'F1=', F1, 'F2=', F2
  STOP
  END
```

س٣/ اكتب برنامجاً لحساب المشتقة التفاضلية الاولى والثانية للاقتران:
$f(x)=x^2 \sin x$ ، ثم جد قيمتيهما عند $x=\frac{\pi}{2}$.

ج/ اكتب برنامجاً لحساب المشتقتين التفاضليتين الاولى والثانية للاقتران: $f(x)=\frac{x^3+2x+3}{x^2+4}$ ، عندما $x=-2$.

7.4 التكامل العددي Numerical Integration

a. قاعدة شبه المنحرف Trapezoidal Rule

لإيجاد القيمة التقريبية للتكامل:

$$\int_a^b f(x)\,dx$$

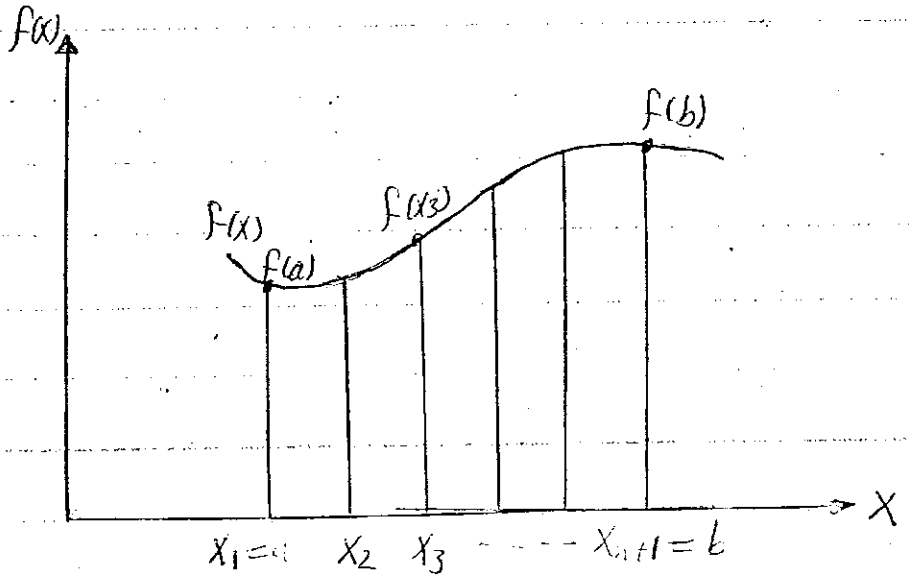
باستخدام قاعدة شبه المنحرف، فإننا نتبع الخطوات الآتية:

1- نجزئ الفترة من a الى b إلى N من الفترات الجزئية، متساوية الأطوال، طول كل منها يساوي ΔX، حيث تحسب ΔX من القاعدة:

$$\Delta X = \frac{b-a}{N}$$

والشكل التالي يوضح هذه التجزئة، حيث ينتج N من الشرائح تحت المنحنى $f(x)$ كل شريحة منها تأخذ شكلاً قريباً من شبه المنحرف، والتي يمكن حساب مساحتها، وفقاً لقاعدة شبه المنحرف:

$$A_i = \left[\frac{f(x_i)+f(x_{i+1})}{2}\right]\Delta X$$



2- لحساب المساحة الكلية تقريبية لطول تحت المنحنى $f(x)$ فإننا نجمع مساحات الشرائح كما يلي:

$$\text{Total Area} = \int_a^b f(x)\,dx = A_1 + A_2 + \cdots + A_N$$

$$= \left[\frac{f(x_1)+f(x_2)}{2}\right]\Delta x + \left[\frac{f(x_2)+f(x_3)}{2}\right]\Delta x + \cdots +$$

$$\left[\frac{f(x_n)+f(x_{n+1})}{2}\right]\Delta x$$

$$= \frac{\Delta x}{2}\left[f(a) + 2\sum_{i=2}^{N} f(x_i) + f(b)\right]$$

$$= \frac{\Delta x}{2}\left[f(a)+f(b)+2\sum_{i=2}^{N} f(x_i)\right]$$

مثال / اكتب برنامجاً لايجاد القيمة التقريبية للتكامل :

$$\int_1^5 \frac{\sin^2 x\, dx}{x}$$

باستخدام قاعدة شبه المنحرف ($N=64$).

```
F(X) = SIN(X) **2 / X
READ *, A, B, N
DX = (B-A) / FLOAT(N)
SUM = F(A) + F(B)
X = A + DX
DO 20, I = 2, N
SUM = SUM + 2.0 * F(X)
20 X = X + DX
SUM = SUM * DX / 2.0
PRINT *, 'INTEGRAL = ', SUM
STOP
END
```

Simpsons Rule

في هذه الطريقة، نقسم الفترة بين a و b الى عدد زوجي N من الفترات الجزئية المتساوية، طول كل منها يساوي Δx،

حيث :

$$\Delta X = \frac{b-a}{N}$$

وفقاً لقاعدة سمبسون فإن قيمة التكامل التقريبية تستخرج من العلاقة التالية :

$$\int_a^b f(x)\,dx = \text{Total Area}$$

$$= \frac{\Delta X}{3}\left[f(X_1) + 4f(X_2) + 4f(X_4) + 2f(X_5) + \cdots + f(X_{n+1})\right]$$

$$= \frac{\Delta X}{3}\left[f(a) + 4\sum_{i=2,4,6}^{N} f(X_i) + 2\sum_{i=3,5,7}^{N-1} f(X_i) + f(b)\right]$$

$$= f(a) + 4f(a+\Delta X) + 2f(a+2\Delta X) + 4f(a+3\Delta X) + \cdots + f(b)$$

مثال / اكتب برنامجاً لإيجاد القيمة التقريبية للتكامل $\int_1^5 \frac{\sin^2 X}{X}\,dX$ ، باستخدام قاعدة سمبسون ($N=64$).

```
F(X) = SIN(X)**2.0/X
READ*, A, B, N
DX = (B-A)/FLOAT(N)
SUM = F(A) + F(B) + 4.0*F(A+DX)
NM1 = N-1
X = A+DX
DO 20, I=2, NM1, 2
SUM = SUM + 2.0*F(X) + 4.0*F(X+DX)
20  X = X+2*DX
SUM = SUM*DX/3.0
PRINT*, 'INTEGRAL =', SUM
```

STOP

END

س/ باستخدام قاعدة شبه المنحرف، اكتب برنامجاً بلغة فورتران لإيجاد القيمة التقريبية لكل من التكاملات الآتية:

a. $\int_{2}^{10} \frac{dx}{\sqrt{x}}$

b. $\int_{1}^{5} x \ln x \, dx$

س/ باستخدام قاعدة سيمبسون، اكتب برنامجاً بلغة فورتران لإيجاد القيمة التقريبية لكل من التكاملات الآتية:

a. $\int_{0}^{2} \frac{dx}{x^3 - 3x - 3}$

b. $\int_{0}^{\pi/4} x \sin 2x \, dx$

5-7 الحل العددي للمعادلات التفاضلية

a- طريقة أويلر Eulers Method

المعادلة التفاضلية $\frac{dy}{dx} = F(x,y)$ معرفة في الفترة $a \le x \le b$ حيث تكون قيمة y معروفة عند نقطة البداية $x=a$، ولتكن هذه القيمة أي $y(a)=c$، والمطلوب هو إيجاد الحل y حيث يكون اقترانا للمتغير x.

طريقة الحل: نجزئ فترة الحل من $x=a$ إلى $x=b$ الى N من الفترات الجزئية المتساوية طول كل منها Δx حيث:

$$\Delta x = \frac{(b-a)}{N}$$

خوارزمية أويلر:

تتكون خوارزمية أويلر لحل المعادلات التفاضلية من الخطوات الآتية:

الخطوة (1) نقرأ قيم الثوابت العددية a، b، c، N

ونضع $\Delta x = (b-a)/N$ ونعرف الاقتران الداخلي $F(x,y)$

ونضع قيمة x الاولى التي يبدأ بها الحل وقيمة y عندها

الخطوة (2) نضع قيمة $i = 1$

الخطوة (3) نضع قيمة x_i بدلالة النقطة التي قبلها:

$x_i = x_1 + (i-1)\Delta x$، ونحسب y_i من العلاقة:

$y_i = y_{i-1} + \Delta x\, F(x_{i-1}, y_{i-1})$، ثم نطبع النتيجة: x_i, y_i

الخطوة (4) اذا كان i أصغر من N نزد الخطوة

واذا كان $i = N$ نتوقف.

مثال / أوجد الحل التقريبي لمعادلة تفاضلية

$$\frac{dy}{dx} = -y + x + 1$$

حيث : $0 \leq X \leq 1$

$y(0) = 1$ بطريقة أويلر

```
C     EULER'S METHODS
      DIMENSION X(11), Y(11).
      READ*, A, B, C, N .
      F(X,Y) = -Y + X + 1.0
      DX = (B-A)/N
      X(1) = A
      Y(1) = C
      PRINT*, 'FOR X=', X(1), 'Y=', Y(1)
      I = 2
3     X(I) = X(1) + (I-1)*DX
      Y(I) = Y(I-1) + DX * F(X(I-1), Y(I-1))
      PRINT*, 'FOR X=', X(I), 'Y=', Y(I)
      IF (I.GE.(N+1)) GoTo 5
      I = I+1
      GoTo 3
5     STOP
      END
```

مثال / جد الحل التقريبي للمعادلة التفاضلية

$\frac{dy}{dx} = y^2 + x$ عند $x = 2$

حيث $y(1) = 1$ ، ستعمل طريقة أويلر وبحيث تكون $\Delta x = 0.1$

6- طريقة رانج كتا ذات الرتبة الرابعة Fourth-Order Runge-Kutta Method

لحل المعادلة $dy/dx = F(x,y)$ في الفترة $a \le x \le b$ حيث $y(a) = c$

نجزئ المسافة بين $x=a$ و $x=b$ الى N من الفترات الجزئية المتساوية طول كل منها Δx ثم نجد قيمة y من المعادلة:

$$y_{i+1} = y_i + \frac{1}{6}[K_1 + 2K_2 + 2K_3 + K_4]$$

حيث كل من: K_1, K_2, K_3, K_4 هي ثوابت.

خوارزمية الحل:

الخطوه (1) نعرّف $F(x,y)$

ونقرأ قيم الثوابت a, b, c, N

ونضع قيمة $\Delta x = (b-a)/N$ ، ثم نضع قيمتي X_1 و Y_1

الخطوه (2) نضع قيمة $i=1$

الخطوه (3) نحسب قيم الثوابت K_1, K_2, K_3, K_4 لهذه الخطوة من العلاقات الآتية:

$$K_1 = \Delta X \; F(X_i, y_i)$$

$$K_2 = \Delta X \; F\left(X_i + \frac{\Delta X}{2}, y_i + \frac{K_1}{2}\right)$$

$$K_3 = \Delta X \; F\left(X_i + \frac{\Delta X}{2}, y_i + \frac{K_2}{2}\right)$$

$$K_4 = \Delta X \; F(X_i + \Delta X, y_i + K_3)$$

ثم نحسب y_{i+1} من العلاقة:

$$y_{i+1} = y_i + \frac{1}{6}[K_1 + 2K_2 + 2K_3 + K_4]$$

ونحسب X_{i+1} من العلاقة:

$$X_{i+1} = X_i + \Delta X$$

ثم نطبع قيم كل من X_{i+1} و y_{i+1}

الخطوه (4) إذا كانت $i \ge N$ فاذهب الى الخطوة 6.

الخطوه (5) أضف 1 الى i ثم عد الى الخطوه 3.
الخطوه (6) توقف.

مثال/ أوجد الحل التقريبي العددي للمعادلة التفاضلية:

$$\frac{dy}{dx} = -y + x + 1$$

حيث: $0 \leq x \leq 1$

$y(0) = 1$ بطريقة رانج - كتا

```
DIMENSION X(11), Y(11)
READ*, A, B, C, N
F(X, Y) = -Y + X + 1
DX = (B-A) /N
X(1) = A
Y(1) = C
PRINT*, 'FOR X = ', X(1), 'Y = ', Y(1)
I = 1
3  AK1 = DX * F(X(I), Y(I))
   Ak2 = DX * F(X(I) + DX/2., Y(I) + AK1/2.0)
   Ak3 = DX * F(X(I) + DX/2.0, Y(I) + Ak2/2.0)
   AK4 = DX * F(X(I) + DX, Y(I) + Ak3)
   Y(I+1) = Y(I) + (AK1 + 2.0 * AK2 + 2. * Ak3 + Ak4)/6.
   X(I+1) = X(1) + I * DX
   PRINT*, 'FOR X = ', X(I+1), 'Y = ', Y(I+1)
   IF (I.GE.N) GO TO 6
   I = I + 1
   GO TO 3
6  STOP
   END
```

س/ استخدم طريقة رانج كتا، لحل المعادلة التفاضلية:

$$\frac{dy}{dt} = 1 + t \sin(ty)$$

حيث: $0 \leq t \leq 2$ ، $y(0)=0$ ، $\Delta x = 0.1$

س/ استخدم طريقة أويلر ($\Delta x = 0.1$) لايجاد الحل التقريبي للمعادلة التفاضلية:

$$\frac{dy}{dt} = 2\frac{y}{t} + t^2 e^t \qquad 1 \leq t \leq 2 \; ; \; y(1)=0$$

أحل المسألة باستخدام طريقة رانج كتا.