الفصل الثامن

الحلول العددية
للمعادلات التفاضلية العادية

يحتوي هذا الفصل على:

1.8 مقدمة

مما لا شك فيه أن القارئ سبق له وأن تعرض لدراسة المعادلات التفاضلية العادية وحلولها التحليلية. فعلى سبيل المثال نرى أن المعادلة:

$$\frac{d^2 y}{dx^2}+3\frac{dy}{dx}-15y=0$$

معادلة تفاضلية عادية متجانسة وخطية؛ بينما المعادلة:

$$\frac{d^2 y}{dx^2}+y\frac{dy}{dx}-7x=0$$

هي معادلة تفاضلية عادية غير متجانسة وغير خطية.

في دراستنا للحلول العددية للمعادلات التفاضلية العادية سوف نتعرض في البداية للمعادلات من النوع :

$$\frac{dy}{dx}= f(x,y) \qquad \ldots\ldots (1.8)$$

أي لمعادلة من الرتبة الأولى وحيث $f(x,y)$ هي دالة في x و y .

عموما وإذا استطعنا حل المعادلة (1.8) فإننا نقول بأننا حصلنا على حل تحليلي. فعلى سبيل المثال، أحد الحلول التحليلية للمعادلة: $\frac{dy}{dx}=3x^2$ هو $y=x^3$. وإذا أردنا إيجاد قيمة معينة لـ y مماثلة لقيمة ما (x) فإننا نستعمل الحاسبات الآلية لإيجاد الحل (وذلك عندما يكون الحل معقد التركيب).

كما نعلم ونحن بصدد إيجاد حل المعادلة التفاضلية ربما نواجه عدداً لا نهائياً من الحلول، ولكي نحصل على حل واحد فقط يجب أن نعين بعض الشروط مثل الشروط

الابتدائية أو الشروط الحدية. مثلاً أن نذكر بأن المنحنى يمر بالنقطة (x_o, y_o) حيث (x_o, y_o) هي نقطة ما .

ففي المثال السابق يكون الحل العام هو $y = x^3 + c$ وإذا اشترطنا أن $y(1) = 2$ فإن $c = 1$ ويكون الحل (الوحيد) هو: $y = x^3 + 1$.

وباستعمال نظريات المعادلات التفاضلية نجد أنه بمجرد تعيين الشروط الكافية نحصل على حل واحد لا غير للمعادلة.

لإيجاد الحلول العددية للمعادلات التفاضلية العادية نورد بعضاً من الطرق المستخدمة لهذا الغرض، فيما يلي:

2.8 طريقة أويلر (Euler's Method (EM))

لنفترض أن $\frac{dy}{dx} = f(x, y)$ وأن $y = y_o$ عند $x = x_o$.

إذا كان من الممكن الحصول على الحل تحليلياً كان بها، وإذا تعذر ذلك فإننا نستعمل الطرق العددية، وأبسط هذه الطرق هي طريقة أويلر.

ولتوضيح استخدام الطريقة العددية يمكننا تصوير الآتي: ((افترض أننا نبحث عن كنز مخبأ؛ إذا حصلنا على الحل التحليلي فذلك يكون بمثابة حصولنا على خريطة جاهزة للكنز. بينما يكون الحل العددي بمثابة حصولنا على نقطة البداية وفئة من التوجيهات عن طريقها نتبع الطريق الذي يؤدي إلى الهدف المنشود)).

كما ذكرنا سابقاً لنفترض أن نقطة البداية هي (x_o, y_o) أي أن:

$$\left.\frac{dy}{dx}\right|_{(x_o,y_o)} = f(x_o, y_o)$$

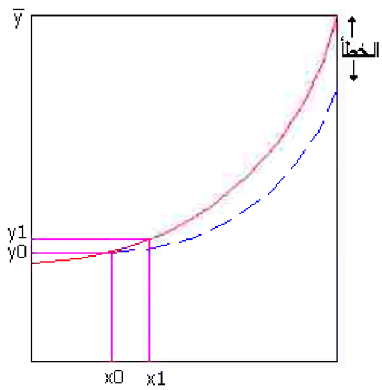
ولنفترض أيضاً أن الحل $y = F(x)$ مستمر وقابل للتفاضل. عندئذ ولو كان الحل مطلوباً عند $x = \bar{x}$ ،أي أنه علينا تعيين $y = F(\bar{x})$ فإننا نجزئ الفترة $[x_o, \bar{x}]$ إلى n من الفترات الفرعية بعرض قدره ω و حيث:

$$\omega = \frac{\bar{x} - x_o}{n} \qquad \ldots\ldots (2.8)$$

وبهذا تكون النقاط الحدودية هي:

$$x_o, x_1, x_2, \ldots\ldots, x_n (= \bar{x})$$

أنظر الشكل (1.8) الموضح أسفله.



الشكل (1.8) توضيح للحل العددي للمعادلات التفاضلية

ويتضح من الشكل (1.8) ،وكتقريب أولي، أن:

$$\left.\frac{dy}{dx}\right|_{(x_o,y_o)} = f(x_o, y_o) = \frac{y_1 - y_o}{x_1 - x_o} \qquad \ldots\ldots (3.8)$$

أو أن:

$$y_1 = y_o + \omega\ f(x_o, y_o) \qquad \ldots\ldots (4.8)$$

(يمكن أيضاً الحصول على هذه النتيجة باستخدام مفكوك تايلور كما سنوضحه فيما بعد).

بنفس الطريقة نحصل على:

$$\begin{aligned} y_2 &= y_1 + f(x_1, y_1)\omega \\ &\vdots \\ y_n &= y_{n-1} + f(x_{n-1}, y_{n-1})\omega \end{aligned}$$

وحيث تكون ω صغيرة.

لاحظ الآتي:

1. لكي يكون الخطأ أقل ما يمكن علينا أن نختار ω صغيرة.
2. ازدياد عدد الفترات الفرعية يقود إلى حسابات كثيرة وربما أدى ذلك إلى الوقوع في الخطأ.
3. كل y_i تعتمد على التي قبلها وعليه يجب أن نولي الأمر عناية فائقة وإلا وقعنا في الكثير من الأخطاء بل و لأنتشرت في كل حساباتنا.
4. إذا حدث خطأ ولم يسبب في أخطاء أخرى فإننا نسمي العملية بالمتزنة، بينما تكون العملية غير متزنة إذا حدث غير ذلك.

3.8 امتداد طريقة أويلر ((Extended Euler Method (EEM))

من طريقة أويلر البسيطة وجدنا أن:

$$y_{i+1} = y_i + f(x_i, y_i)(x_{i+1} - x_i) \qquad \ldots\ldots (5.8)$$

هذا يمكن مقارنته بمفكوك متسلسلة تايلور:

$$F(x_{i+1}) = F(x_i) + F'(x_i)\,\Delta x_i \qquad \ldots\ldots (6.8)$$

غير أنه لو شملنا ثلاثة حدود من متسلسلة تايلور فإنه يكون لدينا

$$F(x_{i+1}) \cong F(x_i) + F'(x_i)\,\Delta x_i + F''(x_i)\,\frac{(\Delta x_i)^2}{2!}$$

وبالمقارنة نحصل على:

$$y_{i+1} = y_i + f(x_i, y_i)\omega + \frac{d}{dx}(f(x_i, y_i))\frac{\omega^2}{2} \qquad \ldots\ldots (7.8)$$

والمعادلة (7.8) هي ما نسميها بامتداد طريقة أويلر وتضفي هذه دقة أكثر.

نلاحظ أن:

$$\frac{d}{dx}f(x, y) = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\cdot\frac{dy}{dx} = \frac{\partial f}{\partial x} + f\frac{\partial f}{\partial y} \qquad \ldots\ldots (8.8)$$

4.8 طريقة أويلر الأكثر امتداد (More Extended Euler Method (MEEM))

لو استمرينا في إضافة حدود أخرى من مفكوك متسلسلة تايلور وشملنا الحدود الأربعة الأولى أي أننا اعتبرنا:

$$F(x_{i+1}) \cong F(x_i) + F'(x_i)\,\Delta x_i + F''(x_i)\,\frac{(\Delta x_i)^2}{2!} + F'''(x_i)\,\frac{(\Delta x_i)^3}{3!}$$

فإنه بالمقارنة نحصل علي التكرار:

$$y_{i+1} = y_i + f(x_i, y_i)\omega + \frac{d}{dx}\left(f(x_i, y_i)\right)\frac{\omega^2}{2} + \frac{d^2}{dx^2}\left(f(x_i, y_i)\right)\frac{\omega^3}{6} \qquad \ldots\ldots (9.8)$$

وحيث نرى أن:

$$\begin{aligned}\frac{d^2 f(x,y)}{dx^2} &= \frac{d}{dx}\left(\frac{\partial f}{\partial x} + f\frac{\partial f}{\partial y}\right)\\ &= \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x} + f\frac{\partial f}{\partial y}\right) + f\frac{\partial}{\partial y}\left(\frac{\partial f}{\partial x} + f\frac{\partial f}{\partial y}\right) \qquad \ldots\ldots (10.8)\\ &= \frac{\partial^2 f}{\partial x^2} + 2f\frac{\partial^2 f}{\partial y \partial x} + \frac{\partial f}{\partial x}\cdot\frac{\partial f}{\partial y} + f\left(\frac{\partial f}{\partial y}\right)^2 + f^2\frac{\partial^2 f}{\partial y^2}\end{aligned}$$

وتعتبر المعادلة (9.8) هي طريقة أويلر الأكثر امتدادا.

فيما يلي نورد عدة أمثلة، نطبق فيها الطرق الثلاث ومن ثم نقارن بينها.

مثال (1.8)

لنأخذ في الاعتبار المعادلة: $\frac{dy}{dx} = y \quad ; \quad y(0) = 1$

و ليكن المطلوب هو إيجاد قيمة y عند $\bar{x} = 1$.

من الواضح أن الحل التحليلي لهذا المثال هو $y = e^x$ وما نحن بصدد حسابه أو

معرفته هو العدد $e^1 = 2.718281$ ولو رمزنا للطرق الثلاث بـ EM و EEM و $MEEM$ على التوالي فإن التكرارات للطرق الثلاث هي:

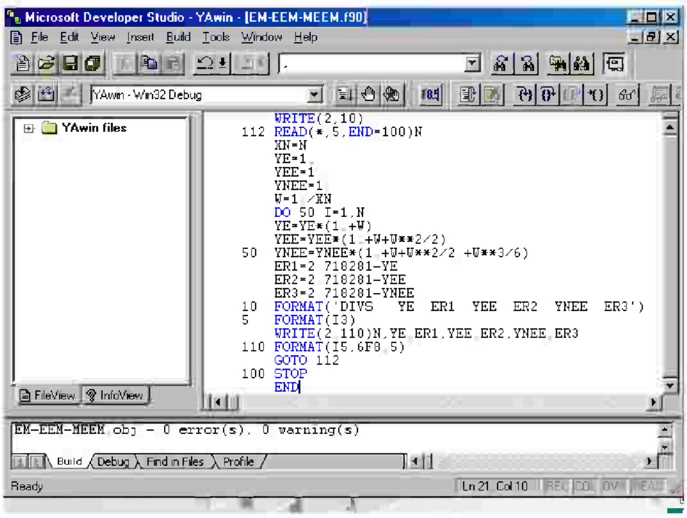
$$EM \quad : \quad y_{i+1} = y_i(1+\omega)$$

$$EEM \quad : \quad y_{i+1} = y_i\left(1+\omega+\frac{1}{2}\omega^2\right)$$

$$MEEM \quad : \quad y_{i+1} = y_i\left(1+\omega+\frac{1}{2}\omega^2+\frac{1}{6}\omega^3\right)$$

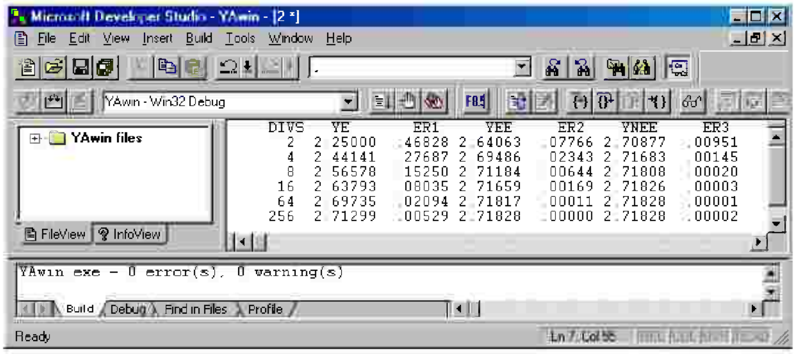
(وحيث $\omega = \dfrac{1}{N}$ و N هو عدد الفترات)

وبكتابة البرنامج الموضح بالشكل (2.8) نحصل على النتائج الموضحة بالجدول (1.8). ومنه نلاحظ أن طريقة أويلر الأكثر امتدادا أعطت نتائج قريبة من القيمة المتوقعة حتى لفترتين فرعيتين وهو ما نتوقعه من هذه الطريقة من توفير للجهد والوقت ومن حصول على دقة أكبر.



الشكل (2.8) حل المعادلة التفاضلية $\frac{dy}{dx} = y,\ y(0)=1$ باستخدام الطرق الثلاث $YE(EM), YEE(EEM), YNEE(MEEM)$.

الجدول (1.8) نتائج الطرق الثلاثة $(EM),(EEM),(MEEM)$ للمثال $y=e^x$.



DIVS	YE	ER1	YEE	ER2	YNEE	ER3
2	2.25000	.46828	2.64063	.07766	2.70877	.00951
4	2.44141	.27687	2.69486	.02343	2.71683	.00145
8	2.56578	.15250	2.71184	.00644	2.71808	.00020
16	2.63793	.08035	2.71659	.00169	2.71826	.00003
64	2.69735	.02094	2.71817	.00011	2.71828	.00001
256	2.71299	.00529	2.71828	.00000	2.71828	.00002

$DIVS$ تمثل عدد الفترات و $YE, YEE, YNEE$ الحلول و $ER2, ER1$ و $ER3$ الأخطاء.

مثال (2.8)

في هذه الحالة نأخذ المعادلة: $y(1)=1 \quad , \quad \frac{dy}{dx}=-y^2$ ونوجد الحل عند $\bar{x}=2$.

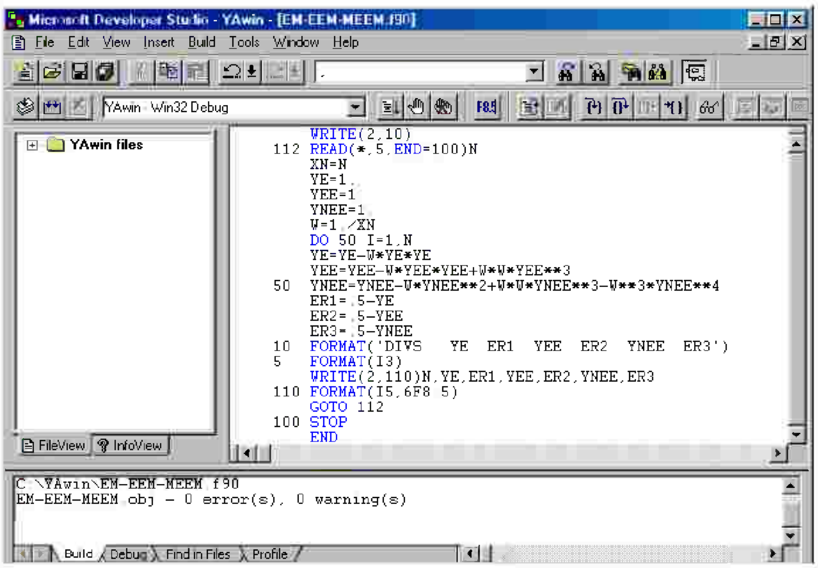
نرى وبكل وضوح أن الحل هنا هو $y=\frac{1}{x}$ وهكذا تكون القيمة المطلوبة هي 0.5.

وباستخدام الطرق العددية الثلاث نرى أن:

$$EM \quad : \quad y_{i+1} = y_i - \omega\, y_i^2$$
$$EEM \quad : \quad y_{i+1} = y_i - \omega\, y_i^2 + \omega^2\, y_i^3$$
$$MEEM \quad : \quad y_{i+1} = y_i - \omega\, y_i^2 + \omega^2\, y_i^3 - \omega^3\, y_i^4$$

($\omega=\frac{1}{N}$ و N هو عدد الفترات الفرعية)

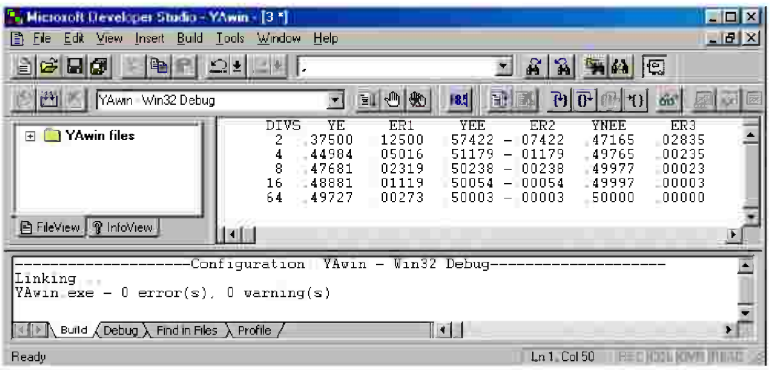
بكتابة البرنامج الموضح بالشكل (3.8) نحصل على النتائج الموضحة بالجدول (2.8)



```
      WRITE(2,10)
112   READ(*,5,END=100)N
      XN=N
      YE=1.
      YEE=1
      YNEE=1
      W=1./XN
      DO 50 I=1,N
      YE=YE-W*YE*YE
      YEE=YEE-W*YEE*YEE+W*W*YEE**3
50    YNEE=YNEE-W*YNEE**2+W*W*YNEE**3-W**3*YNEE**4
      ER1=.5-YE
      ER2=.5-YEE
      ER3=.5-YNEE
10    FORMAT('DIVS    YE   ER1   YEE   ER2   YNEE   ER3')
5     FORMAT(I3)
      WRITE(2,110)N,YE,ER1,YEE,ER2,YNEE,ER3
110   FORMAT(I5,6F8.5)
      GOTO 112
100   STOP
      END
```


الشكل (3.8) برنامج لحل المعادلة التفاضلية العادية $\frac{dy}{dx}=-y^2,\ y(1)=1$ عند $\bar{x}=2$ باستخدام الطرق الثلاث $(EM),(EEM),(MEEM)$.

الجدول (2.8) نتائج المثال (2.8) $y=\frac{1}{x}$.



DIVS	YE	ER1	YEE	ER2	YNEE	ER3
2	.37500	.12500	.57422	-.07422	.47165	.02835
4	.44984	.05016	.51179	-.01179	.49765	.00235
8	.47681	.02319	.50238	-.00238	.49977	.00023
16	.48881	.01119	.50054	-.00054	.49997	.00003
64	.49727	.00273	.50003	-.00003	.50000	.00000

$DIVS$ عدد الفترات الفرعية.

$YE, YEE, YNEE$ هي القيم بدلالة الطرق الثلاث. $ER1, ER2, ER3$ الأخطاء.

مثال (3.8)

في هذا المثال نأخذ في الحسبان المعادلة: $y(1)=1 \quad , \quad \frac{dy}{dx}=3x^2$

والمطلوب إيجاد الحل عند $\bar{x}=2$.

نرى أن الحل التحليلي هو x^3 ، وهو يحقق الشرط الابتدائي وأنه باستعمال الطرق الثلاث نحصل على:

$$\begin{aligned}
&EM \quad : \quad y_{i+1} = y_i + 3\omega\, x_i^2 \\
&EEM \quad : \quad y_{i+1} = y_i + 3\omega\, x_i^2 + 3\omega^2\, x_i^3 \\
&MEEM \quad : \quad y_{i+1} = y_i + 3\omega\, x_i^2 + 3\omega^2\, x_i^3 + \omega^3
\end{aligned}$$

($\omega=\frac{1}{N}$ و N هو عدد الفترات الفرعية)

مرة أخرى نقوم بكتابة البرنامج الموضح بالشكل (4.8) لإجراء الحسابات المطلوبة ونبرز النتائج بالجدول (3.8).

لاحظ من الجداول (3.8-1.8) أن:

$DIVS$ هو عدد الفترات الفرعية المستعملة.

YE هي القيمة المحسوبة بطريقة أويلر EM .

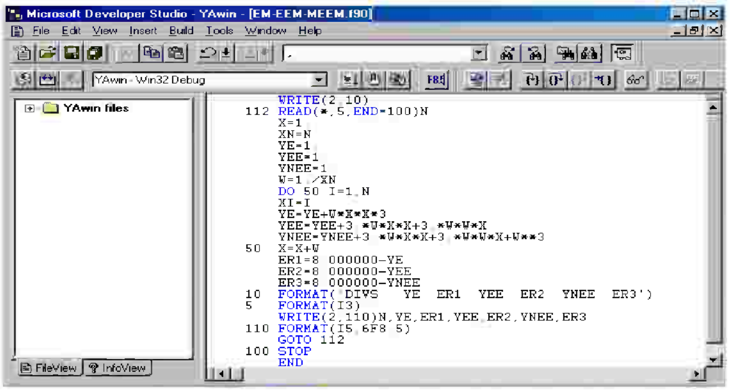
YEE هي القيمة المحسوبة بطريقة أويلر EEM .

$YNEE$ هي القيمة المحسوبة بطريقة أويلر $MEEM$.

$ER1$ هو الخطأ في القيمة المحسوبة بطريقة EM .

$ER2$ هو الخطأ في القيمة المحسوبة بطريقة EEM .

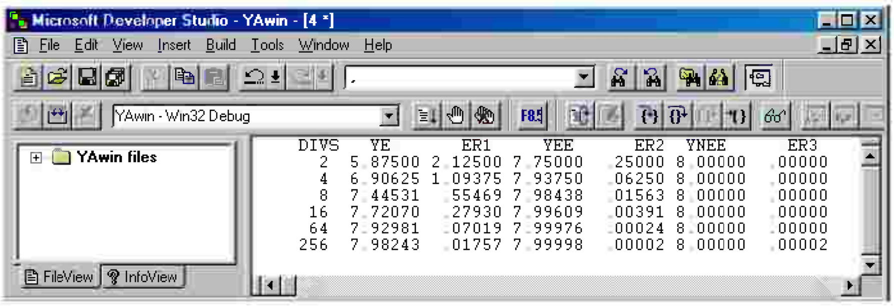
$ER3$ هو الخطأ في القيمة المحسوبة بطريقة $MEEM$.




```
      WRITE(2,10)
112   READ(*,5,END=100)N
      X=1
      XN=N
      YE=1
      YEE=1
      YNEE=1
      W=1./XN
      DO 50 I=1,N
      XI=I
      YE=YE+W*X*X*3
      YEE=YEE+3.*W*X*X+3.*W*W*X
      YNEE=YNEE+3.*W*X*X+3.*W*W*X+W**3
50    X=X+W
      ER1=8.000000-YE
      ER2=8.000000-YEE
      ER3=8.000000-YNEE
10    FORMAT(' DIVS    YE   ER1   YEE   ER2   YNEE   ER3')
5     FORMAT(I3)
      WRITE(2,110)N,YE,ER1,YEE,ER2,YNEE,ER3
110   FORMAT(I5,6F8.5)
      GOTO 112
100   STOP
      END
```

الشكل **(4.8)** حل المعادلة التفاضلية $\frac{dy}{dx}=3x^2, y(1)=1$ باستخدام الطرق الثلاث $(EM),(EEM),(MEEM)$.

الجدول **(3.8)** نتائج البرنامج بالشكل **(4.8)**



DIVS	YE	ER1	YEE	ER2	YNEE	ER3
2	5.87500	2.12500	7.75000	.25000	8.00000	.00000
4	6.90625	1.09375	7.93750	.06250	8.00000	.00000
8	7.44531	.55469	7.98438	.01563	8.00000	.00000
16	7.72070	.27930	7.99609	.00391	8.00000	.00000
64	7.92981	.07019	7.99976	.00024	8.00000	.00000
256	7.98243	.01757	7.99998	.00002	8.00000	.00002

$DIVS$ هو عدد الفترات الفرعية.

$YE, YEE, YNEE$ القيم للطرق الثلاث.

$ER1, ER2, ER3$ الأخطاء.

لاحظ أيضاً أن عدد الفترات التي تم الحساب بها هي:

$$N = 2,4,8,16,64,256$$

كما يتضح من هذه الأمثلة أنه لمثل هذه الحالات يكفي أن نأخذ عدد الفترات على أنه 256 كحد أقصى، غير أنه يمكن أخذ عدد الفترات حسب الدقة التي ننشدها. كما ذكرنا، آنفاً، لقد تميزت طريقة أويلر الأكثر امتداداً بأنه من خلالها يمكن اعتبار عدد من الفترات أقل من ذلك المستخدم في طريقتي أويلر وامتدادها.

فعلى سبيل المثال نلاحظ بالأمثلة السابقة أنه للحالتين الأولتين أعطت $N=8$ نتائج طيبة بطريقة أويلر الأكثر امتدادا وفي الحالة الأخيرة ولـ $N=2$ (فقط) أعطت الطريقة القيمة المتوقعة تماما.

وهكذا نلاحظ الدور الهام الذي تلعبه متسلسلة تايلور بخصوص حل المعادلات التفاضلية العادية عددياً؛ ومن خلال إضافة عدد أكبر من الحدود يمكنا الحصول على الدقة المطلوبة.

5.8 طريقة أويلر المعدلة ((Modified Euler Method (MEM))

يطلق على الطرق السابقة كلها بالطرق ذات الخطوة المفردة (Single-Step) وذلك لأنها تستعمل نقطة واحدة في كل خطوة من الحسابات. افترضنا أيضاً في الطرق المذكورة أن المشتقة ثابتة في كل فترة فرعية؛ غير أن هذا ليس صحيحاً بالضبط وعليه نلجأ لطريقة أخرى حيث نتعامل فيها بمتوسط المشتقة في الفترة الفرعية بدلاً من المشتقة عند إحدى النقاط وعليه وفي هذه الطريقة المعدلة نقوم بالآتي:

1. نبدأ بالنقطة (x_i, y_i) ثم نستعمل طريقة أويلر (EM) لحساب $\bar{y}_{i+1}$ التي تماثل x_{i+1} وهذه القيمة تسمى بالمنبئ (Predictor).

2. هذه الخطوة هي خطوة المصحح (Corrector)

3. نستخدم $(x_{i+1}, \bar{y}_{i+1})$ لحساب $f(x_{i+1}, \bar{y}_{i+1})$ ثم نحسب المتوسط للقيمتين $f(x_i, y_i)$ و $f(x_{i+1}, \bar{y}_{i+1})$.

نرجع الآن لـ y_{i+1} ونحسبها ولكن باستعمال المتوسط المحسوب للمشتقة وتكون القيمة المحسوبة هي القيمة المتوقعة.

مثل هذه الطرق تسمى بطرق المنبئ والمصحح. وهذه الطريقة بالذات تسمى بطريقة أويلر المعدلة. فعلى سبيل المثال لو كانت:

$$\frac{dy}{dx} = y = f(x, y), \quad y(0) = 1$$

وأن عدد الفترات الفرعية عبارة عن فترة فرعية واحدة؛ أي أن:

$$\omega = \frac{1-0}{1} = 1$$

فإن:

$$\bar{y}_1 = y_0 + f(x_o, y_o)\,\omega = y_o + y_o = 2$$

نحصل الآن على متوسط المشتقة و ذلك كما يلي:

$$\left.\frac{dy}{dx}\right|_{(x_1, y_1)} = f(x_1, \bar{y}_1) = f(1,2) = 2$$

و بذلك فإن:

$$\frac{dy}{dx}=\frac{2+1}{2}=1.5$$

و منها نرى أن: $y_1 = y_0 + 1.5w = 2.5$

وهي قيمة أقرب للقيمة الصحيحة من القيمة المحسوبة بطريقة أويلر.

وعموماً نستطيع تلخيص طريقة أويلر المعدلة في الخطوات التالية:

1. احسب $\bar{y}_{i+1} = y_i + f(x_i, y_i)\omega$.

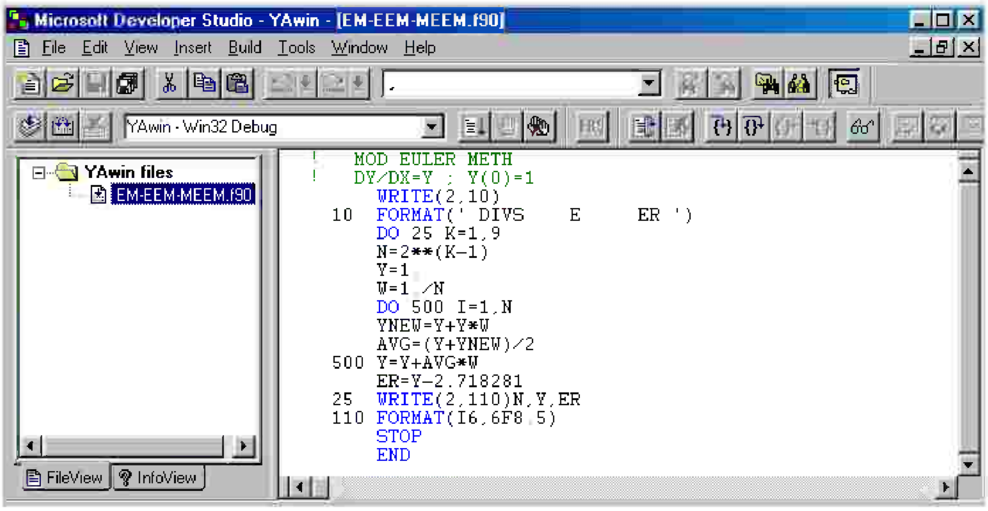
2. احسب $f(x_{i+1}, \bar{y}_{i+1})$ ثم $\frac{1}{2}\left(f(x_{i+1}, \bar{y}_{i+1}) + f(x_i, y_i)\right)$.

3. احسب

$$y_{i+1} = y_i + \frac{w}{2}\left(f(x_i, y_i) + f(x_{i+1}, \bar{y}_{i+1})\right) \qquad \ldots\ldots (11.8)$$

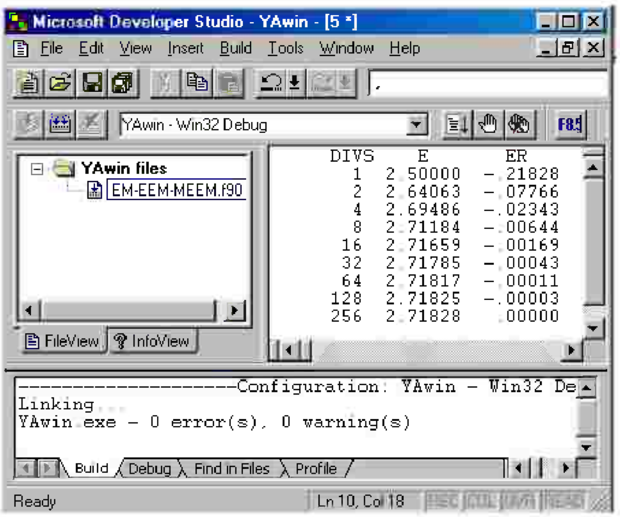
وأعد الكرة.

وباعتبار المثال السابق يمكننا كتابة البرنامج الموضح بالشكل (5.8) والحصول على النتائج الموضحة بالجدول (4.8) ومنها نستطيع تبين مدى الدقة التي أضفتها الطريقة مقارنة بطريقة أويلر حيث نرى أنها أعطت دقة كافية وباستخدام ثماني فترات فرعية (لاحظ أن الدقة المماثلة يمكن الوصول إليها بطريقة أويلر باستخدام 256 فترة فرعية). ولكن مقارنة بامتداد طريقة أويلر نرى أن الدقة، في هذا المثال ، تكاد تكون متقاربة بينما نرى أن طريقة أويلر الأكثر امتداد أدق من الجميع وأسرع تقارباً.



الشكل (5.8) برنامج يقوم بحل المعادلة $\frac{dy}{dx}=y\,,\,y(0)=1$ عند $\bar{x}=1$ وذلك باستخدام طريقة أويلر المعدلة MEM.

الجدول (4.8) نتائج البرنامج بالشكل (5.8)



$DIVS$ عدد الفترات الفرعية.

E القيمة المحسوبة.

ER الخطأ في القيمة المحسوبة.

6.8 طريقة ملن ((Milne's Method (MM)

لو رجعنا للوراء قليلاً لنذكر بأننا استخدمنا طريقة المنبئ والمصحح عندما قمنا بتعديل طريقة أويلر واستخدمنا التكرار:

$$\bar{y}_{i+1} = y_i + f(x_i, y_i)\omega \qquad ...... (12.8)$$

للمنبئ بينما استخدمنا التكرار:

$$y_{i+1} = y_i + \frac{\omega}{2}\left[f(x_i, y_i) + f(x_i, \bar{y}_{i+1})\right] \qquad ...... (13.8)$$

كمصحح.

الآن نعود للعلاقة:

$$\frac{dy}{dx} = \frac{dF(x)}{dx} = f(x, y) \qquad ...... (14.8)$$

ونكاملها من x_{i-1} إلى x_{i+1} وذلك باستخدام طريقة سمبسن لنحصل على:

$$\int_{x_{i-1}}^{x_{i+1}} F'(x)\, dx = \int_{x_{i-1}}^{x_{i+1}} f(x, y)\, dx$$

أو أن:

$$F(x_{i+1}) - F(x_{i-1}) = \frac{\omega}{3}\left[f(x_{i-1}, y_{i-1}) + 4\, f(x_i, y_i) + f(x_{i+1}, y_{i+1}) \right] \qquad \ldots\ldots (15.8)$$

ولو افترضنا أن $y_{i+1} \cong F(x_{i+1}), \quad y_{i-1} \cong F(x_{i-1})$

فإن المصحح يصبح على النحو:

$$y_{i+1} = y_{i-1} + \frac{\omega}{3}\left[f(x_{i-1}, y_{i-1}) + 4\, f(x_i, y_i) + f(x_{i+1}, \bar{y}_{i+1}) \right] \qquad \ldots\ldots (16.8)$$

حيث وضعنا المنبئ $\bar{y}_{i+1}$ إلى اليمين وعلينا إيجاده، ولإيجاد $\bar{y}_{i+1}$ يمكننا استخدام طريقة أويلر ولكن (ملن) يسعفنا بالمنبئ؛ حيث قام باشتقاق المنبئ التالي:

$$\bar{y}_{i+1} = y_{i-3} + \frac{4\omega}{3}\left[2\, f(x_i, y_i) - f(x_{i-1}, y_{i-1}) + 2\, f(x_{i-2}, \bar{y}_{i-2}) \right] \qquad \ldots\ldots (17.8)$$

ومعادلتا المنبئ (17.8) والمصحح (16.8) يكونان طريقة (ملن). وكما هو واضح تتطلب الطريقة معرفة أربع قيم لـ y مسبقاً و هي:

$$y_{i-3}, y_{i-2}, y_{i-1}, y_i$$

وعليه فطريقة (ملن) لا تبدأ تلقائياً بنفسها ولا بد أن يتم استخدام طريقة قبلها للحصول على هذه القيم الأربع.

وهكذا نرى أن هذه الطريقة طريقة متعددة الخطوات مقارنة بطريقة أويلر ذات الخطوة المفردة .لهذه الأسباب نجد أن هذه الطريقة ليست شعبية بالرغم من أنها، وعند إمكانية العمل بها، تعطي نتائج جيدة.

وعند هذه النقطة نود الإشارة إلى أنه توجد ثلاثة مصادر للخطأ وهذه هي:

1. الخطأ الناتج عن الطريقة نفسها ويسمى هذا بخطأ البتر وهذا النوع من الخطأ قابل للتحليل.

2. خطأ التقريب ويصدر هذا عن العمليات الحسابية الكثيرة التي نقوم بها؛ هذا النوع قابل للتحليل أيضاً.

3. خطأ الانتشار ويرجع ذلك إلى أن كل $f(x_i, y_i)$ محسوبة تعتمد على التي قبلها و مثل هذه الأخطاء الأولية لا تتراكم حتى الجواب النهائي فحسب بل وقد تسبب أيضاً في أخطاء أخرى جديدة. وهذا النوع من الخطأ معقد التحليل وصعب التتبع ولمعالجته قدر الإمكان، تستعمل الدقة المضاعفة في الحسابات وكذلك يتم استخدام حاسبات أكبر.

في معظم الحالات نلجأ عادة إلى تخمين خطأ البتر ونحاول أن نصححه. فعلى سبيل المثال يكون الخطأ في القيمة y_i ، بالنسبة لطريقة ملن ،هو:

$$E_i \cong \frac{1}{90} f^{(4)}(x_i, y_i)\,\omega^5 \qquad \ldots\ldots (18.8)$$

والتصحيح في هذه الحالة يسمى بتصحيح هامينج Hamming's.

7.8 طريقة رنج- كوتا ((Runge-Kutta Method (RKM)

حيث إن طريقة أويلر (EM) ذات خطوة مفردة وأن طريقة (ملن) (MM) متعددة الخطوة، فلقد تم توحيد الاثنتين في واحدة وهي طريقة رنج – كوتا (RKM)؛ وتتميز هذه الطريقة بأنها أجود وأدق وبأنها تأخذ نفس مقدار الحسابات التي تأخذها طريقة

أويلر تقريباً. كما أنها أكثر اتزانا من طريقة (ملن) و ذاتية البدء ولا تحتاج لطريقة أخرى لبدئها. بيد أنه يؤخذ على الطريقة أن تحليل الخطأ فيها أقل وضوحاً.

وطريقة رنج – كوتا متعددة الرتب فهي برتبة 2، 3، 4 و5 معتمدة بذلك على عدد الحدود التي يتم اعتبارها من مفكوك تايلور؛ والطريقة الأكثر شعبية هي تلك ذات الرتبة الرابعة؛ وحيث إن برهانها صعب وطويل فإننا سوف نقتصر هنا على سرد النتائج فقط.

لو قمنا بتجزئة $[x_o, \bar{x}]$ إلى n من الفترات الفرعية فعند كل فترة فرعية، ولو استخدمنا طريقة رنج – كوتا، فإننا نقوم بإجراء العمليات التكرارية التالية:

$$y_{i+1} = y_i + \frac{1}{6}\left[k_1 + 2k_2 + 2k_3 + k_4\right] \qquad \ldots\ldots (19.8)$$

حيث:

$$k_1 = f(x_i, y_i)\omega \qquad \ldots\ldots a(20.8)$$

$$k_2 = f\left(x_i + \frac{1}{2}\omega, y_i + \frac{1}{2}k_1\right)\omega \qquad \ldots\ldots b(20.8)$$

$$k_3 = f\left(x_i + \frac{1}{2}\omega, y_i + \frac{1}{2}k_2\right)\omega \qquad \ldots\ldots c(20.8)$$

$$k_4 = f(x_i + \omega, y_i + k_3)\omega \qquad \ldots\ldots d(20.8)$$

لاحظ أن الطريقة تستعمل عدة نقاط وهي تلك ذات الإحداثي السيني:

$$x = x_i\ ,\ x_i + \frac{1}{2}\omega\ ,\ x_i + \omega$$

إلا أنها تكون بنفسها الخطوات الوسط. هذه الطريقة شعبية ودقيقة ومتزنة وسهلة البرمجة على الحاسب الآلي نسبياً.

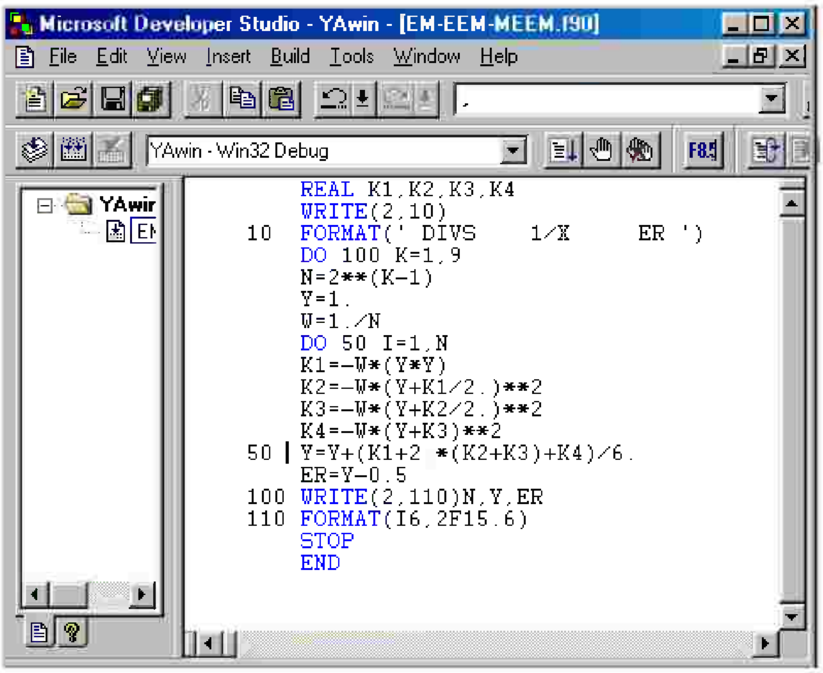
مثال (4.8)

لنأخذ المعادلة:

$$\frac{dy}{dx} = -y^2 \quad ; \quad y(1) = 1$$

والحل مطلوب عند $\bar{x} = 2$.

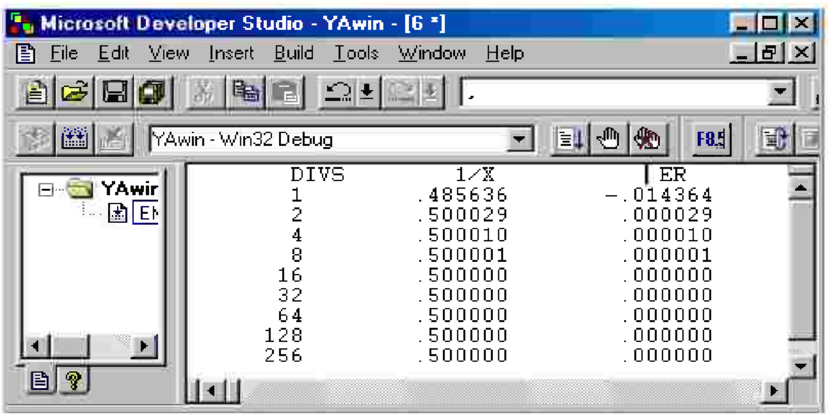
هذا المثال يحتوي على نفس معادلة المثال (2.8). نكتب البرنامج الموضح بالشكل (6.8). لو قمنا بذلك و نفذناه لحصلنا على النتائج الموضحة بالجدول (5.8) والتي من خلالها نرى الدقة المتناهية التي نحصل عليها باستخدام هذه الطريقة (RKM). أي أننا حصلنا على نتائج أفضل مما هي بالطرق السابقة.




```
      REAL K1,K2,K3,K4
      WRITE(2,10)
   10 FORMAT(' DIVS     1/X      ER ')
      DO 100 K=1,9
      N=2**(K-1)
      Y=1.
      W=1./N
      DO 50 I=1,N
      K1=-W*(Y*Y)
      K2=-W*(Y+K1/2.)**2
      K3=-W*(Y+K2/2.)**2
      K4=-W*(Y+K3)**2
   50 Y=Y+(K1+2.*(K2+K3)+K4)/6.
      ER=Y-0.5
  100 WRITE(2,110)N,Y,ER
  110 FORMAT(I6,2F15.6)
      STOP
      END
```

الشكل (6.8) طريقة (RKM) للدالة $y=\frac{1}{x}$.

الجدول (5.8) نتائج البرنامج بالشكل (6.8)



DIVS	1/X	ER
1	.485636	-.014364
2	.500029	.000029
4	.500010	.000010
8	.500001	.000001
16	.500000	.000000
32	.500000	.000000
64	.500000	.000000
128	.500000	.000000
256	.500000	.000000

مثال (5.8)

أوجد حل مسألة القيم الابتدائية $\frac{dy}{dx}=-xy$ $\quad x_o=0\,, y_o=1$ وذلك عند $\bar{x}=1$ بطريقة رنج- كوتا؛ خذ $\omega=0.1$ وقارن بقيم $e^{-x^2/2}$.

الحل:

نكتب البرنامج الموضح بالشكل (7.8) و نقوم بتنفيذه لنحصل على النتائج بالجدول (6.8) ومنها نرى أن :

$$y(1)=0.606531=e^{-1/2}$$

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

      /* STUDENT NAME : ABDELKAREM ALAMEN ABDALLA
         STUDENT NUMBER : 9393
         DEPARTMENT OF CONTROL SYSTEM
         COMPUTER PROGRAM ABOUT NUMERICAL METHODS
         program number (7)Runge Kutta Method*/

void main()
   {
   clrscr();
    int i,n;
float k1,k2,k3,k4,x,y,er,w,y0,x1;
       printf("Enter the number of sup set ");
       scanf("%d",&n);
       printf("The start  point x0=");
       scanf("%f",&x1);
       printf("The start  point y0=");
      scanf("%f",&y0);
         w=.1;
         printf("\nSupset\t\tFunction\tError\t\n");
   for(i=0;i<n;i++)
    {
    k1=-(x*y0)*w;
    k2=-(x+(w/2))*(y0+(k1/2))*w;
    k3=-(x+(w/2))*(y0+(k2/2))*w;
    k4=-(x+w)*(y0+k3)*w;
    y=y0+(k1+2*(k2+k3)+k4)/6;
    er=y-exp(-(w*n)*(w*n)/2);
    printf("%f\t%f\t%f\n",x+.1,y,er);
    x=x+w;
    y0=y;
    }
    getch();
    }
```

الشكل (7.8) – المثال (5.8) بلغة C.

الجدول (6.8) – نتائج المثال (5.8) بلغة C.

```
Enter the number of sup set 10
The start  point x0=0
The start  point y0=1

Supset              Function            Error

0.100000            0.995012            0.388482
0.200000            0.980199            0.373668
0.300000            0.955997            0.349467
0.400000            0.923116            0.316586
0.500000            0.882497            0.275966
0.600000            0.835270            0.228740
0.700000            0.782705            0.176174
0.800000            0.726149            0.119618
0.900000            0.666977            0.060446
1.000000            0.606531            0.000000
```

مثال (6.8)

أكتب برنامجا حاسوبيا يستخدم طريقة رنج كوتا لحل المسألة $\frac{dy}{dx} = \cos x$ بالشرط الابتدائي $F(0) = 0$ وإيجاد قيمة y عند $\pi/2$.

الحل:

البرنامج موضح بالشكل (8.8) بلغة فورتران والنتائج معطاة بالجدول (7.8) وحيث نرى أن:

$$y(\pi/2) = 1$$

```
C    *******      RUNGE -KUTTA METHOD        *****
C    *******  DY/DX=COSX F(0)=0 ,F(90)=...  *****

      REAL K1,K2,K3,K4

      WRITE (5,10)
   10 FORMAT (///'    DIVS    SIN X    ERR'/)

      DO 100 K=1,9
      N =2**(K-1)
      X=0
      W=3.14/(2*N)

      DO 50 I=1,N
      K1=W*(COS(X))
      K2=W*(COS(X+K1/2.))
      K3=W*(COS(X+K2/2.))
      K4=W*(COS(X+K3))
      Y=X+(K1+2.*(K2+K3)+K4)/6.
   50 X=X+W

      ERR=Y-1

  100 WRITE(5,11)N,Y,ERR
   11 FORMAT (7X,I3,4X,F10.6,4X,F10.7)

      STOP
      END
```

الشكل (8.8) – المثال (6.8) بلغة الفورتران.

الجدول (6.8) – نتائج المثال (6.8) بلغة الفورتران.

DIVS	SIN X	ERR
1	1.002275	2.274752E-03
2	1.000134	1.339912E-04
4	1.000008	7.987022E-06
8	1.000000	2.384186E-07
16	9.999998E-01	-1.788139E-07
32	9.999998E-01	-2.384186E-07
64	9.999996E-01	-3.576279E-07
128	9.999991E-01	-8.940697E-07
256	9.999977E-01	-2.324581E-06

والشكل (9.8) يوضح برنامجا بلغة C لنفس المثال (6.8)، أما الجدول (7.8) فيعطي نتائج تنفيذ هذا البرنامج.

```
#include<iostream.h>
#include<iomanip.h>
#include<math.h>
#include<conio.h>
void main()
{clrscr();
int i,j,n;
float k1,k2,k3,k4,y,x,w,er;
cout<<"\n\nw\t\t          y\t\t          er\n\n";
for(i=1;i<=7;i++)
{n=pow(2,i-1);x=0.0;
y=0.0;w=1.570796327/n;
for(j=1;j<=n;j++)
{k1=(cos(x))*w;
k2=(cos(x+(w/2)))*w;
k3=(cos(x+(w/2)))*w;
k4=(cos(x+w))*w;
y=y+((k1+(2*k2)+(2*k3)+k4)/6);
er=y-1.0;
x=x+w;}
cout<<setprecision(6)<<setiosflags(ios::showpoint)<
<w<<"\t\t";
cout<<setprecision(6)<<setiosflags(ios::showpoint)<
<y<<"\t\t";
cout<<setprecision(6)<<setiosflags(ios::showpoint)<
<er<<endl;
}
getch();}
```

الشكل (9.8) – المثال (6.8) بلغة C.

الجدول (7.8) –نتائج المثال (6.8) بلغة C.

y	er
1.002280	0.002280
1.000135	0.000135
1.000008	8.344650e-06
1.000001	5.960464e-07
1.000000	-5.960464e-08
1.000000	-1.788139e-07
1.000000	1.192093e-07

مثال (7.8)

أعد حل المثال $\frac{dy}{dx} = -xy$ $\quad x_o = 0, y_o = 1$ بطريقة رنج–كوتا وطريقة أويلر المعدلة وذلك عند $\bar{x} = 1$ ؛قارن بين الطريقتين.

الحل:

نقوم بحل المسألة بطريقتي رنج-كوتا وأويلر المعدلة فنكتب البرنامج بالشكل (10.8) بلغة باسكال لنحصل بعد تنفيذه على النتائج بالجدول (8.8) وحيث نلاحظ تقارب الحل بالطريقتين وأن الخطأ في طريقة رنج -كوتا أقل. والشكـــــــــلان (11.8) و(12.8) يبين تطابق الحل العددي (●) مع الحل التحليلي (ـــــ).

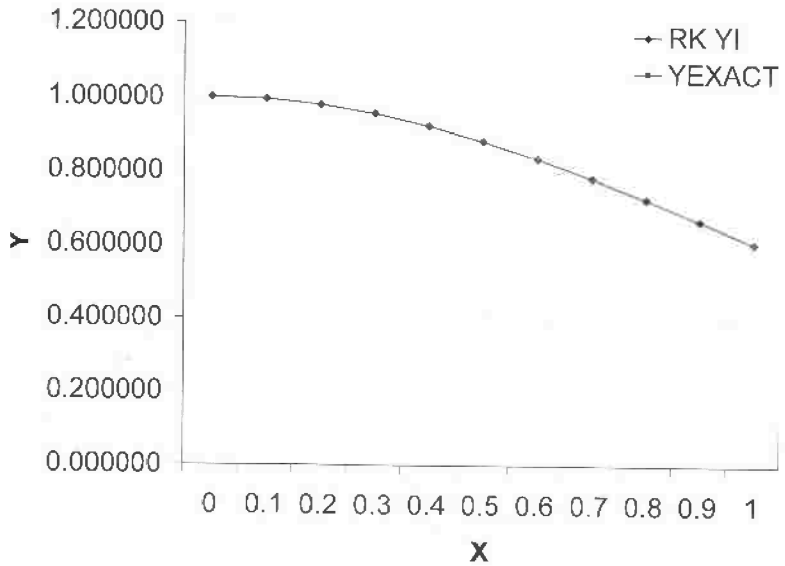
```
DECLARE FUNCTION F! (X!, Y!)
CLS
INPUT "HOW MANY INCREAMENT DO YOU NEED?", N%
INPUT "WHAT IS THE STEP SIZE?", W!
INPUT "WHAT IS THE VALUE OF THE FUNCTION AT THE ZERO POINT", Y!
INPUT "WHAT IS THE STARTING POINT OF THE TIME", X!
Y3! = Y!
Y1! = Y!
Y4! = EXP(-X! ^ 2 / 2)
OPEN "GGZ.TXT" FOR OUTPUT AS #2
OPEN "JRZ.TXT" FOR OUTPUT AS #1
PRINT #2, "X", "MEM Y3", "YEXACT Y4, ER3"
PRINT #2, "-----  ------  -------  --------"
FORMAT$ = "##.##      ##.####^^^^   ##.####^^^^ ###.#####^^^^"
PRINT #2, USING FORMAT$; X!; Y3!; Y4!; ER3
PRINT #1, "X", "RK Y1", "YEXACT Y2", "ERROR"
PRINT #1, "------   -------   --------   -------"

FORMAT$ = "##.##   ##.####^^^^    ##.####^^^^     ##.#####^^^^"
PRINT #1, USING FORMAT$; X!; Y1!; Y2!; ER!
PRINT USING FORMAT$; X!; Y1!; Y2!; ER!

FOR I% = 1 TO N%
Y3! = Y3! + W! * .5 * (F(X!, Y3!) + F(X! + W!, Y3! + W! * F(X!, Y3!)))

K1! = W! * F(X!, Y1!)
K2! = W! * F(X! + .5 * W!, Y1! + .5 * K1!)
K3! = W! * F(X! + .5 * W!, Y1! + .5 * K2!)
K4! = W! * F(X! + W!, Y1! + K3!)
Y1! = Y1! + (K1! + 2 * K2! + 2 * K3! + K4!) / 6

X! = X! + W!

Y4! = EXP(-X! ^ 2 / 2)
ER3! = ABS((Y3! - Y4!) / Y4!) * 100
Y2! = EXP(-X! ^ 2 / 2)
ER! = ABS((Y1! - Y2!) / Y2!) * 100

PRINT #2, USING FORMAT$; X!; Y3!; Y4!; ER3!
PRINT #1, USING FORMAT$; X!; Y1!; Y2!; ER!

NEXT I%
END

FUNCTION F (X!, Y!)
F = -X! * Y!
END FUNCTION
```

الشكل (10.8) المثال (7.8) بلغة باسكال

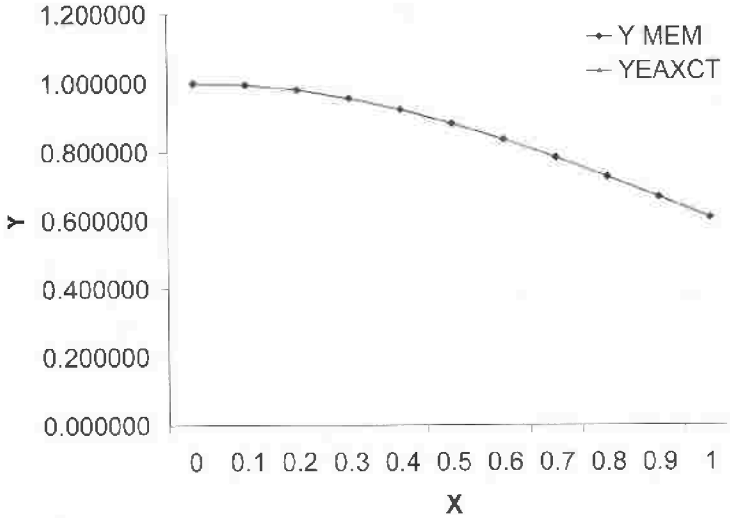
الجدول (8.8) نتائج المثال (7.8) بلغة باسكال

X	RK Y1	YEXACT Y2	ERROR
0.00	1.0000E+00	1.0000E+00	0.00000E+00
0.10	9.9501E-01	9.9501E-01	0.00000E+00
0.20	9.8020E-01	9.8020E-01	0.00000E+00
0.30	9.5600E-01	9.5600E-01	0.00000E+00
0.40	9.2312E-01	9.2312E-01	0.00000E+00
0.50	8.8250E-01	8.8250E-01	0.00000E+00
0.60	8.3527E-01	8.3527E-01	0.00000E+00
0.70	7.8270E-01	7.8270E-01	0.00000E+00
0.80	7.2615E-01	7.2615E-01	0.00000E+00
0.90	6.6698E-01	6.6698E-01	8.93554E-06
1.00	6.0653E-01	6.0653E-01	9.82715E-06

X	MEM Y3	YEXACT Y4	ER3
0.00	1.0000E+00	1.0000E+00	0.00000E+00
0.10	9.9500E-01	9.9501E-01	12.51981E-04
0.20	9.8017E-01	9.8020E-01	24.68835E-04
0.30	9.5596E-01	9.5600E-01	34.85260E-04
0.40	9.2308E-01	9.2312E-01	40.48473E-04
0.50	8.8246E-01	8.8250E-01	37.89045E-04
0.60	8.3525E-01	8.3527E-01	22.26423E-04
0.70	7.8271E-01	7.8270E-01	12.48896E-04
0.80	7.2620E-01	7.2615E-01	73.38239E-04
0.90	6.6709E-01	6.6698E-01	16.89900E-03
1.00	6.0672E-01	6.0653E-01	30.87689E-03

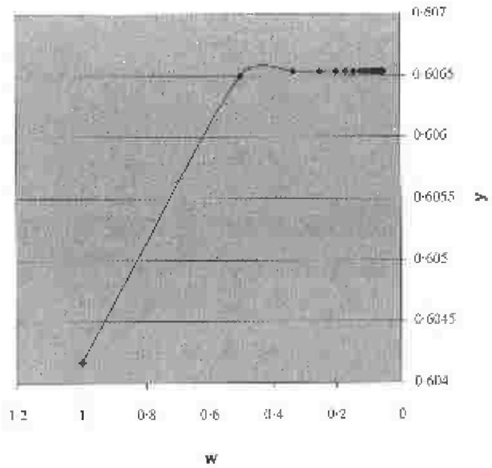


الشكل (11.8) مقارنة مع الحل التحليلي للمثال (7.8) [رنج-كوتا]



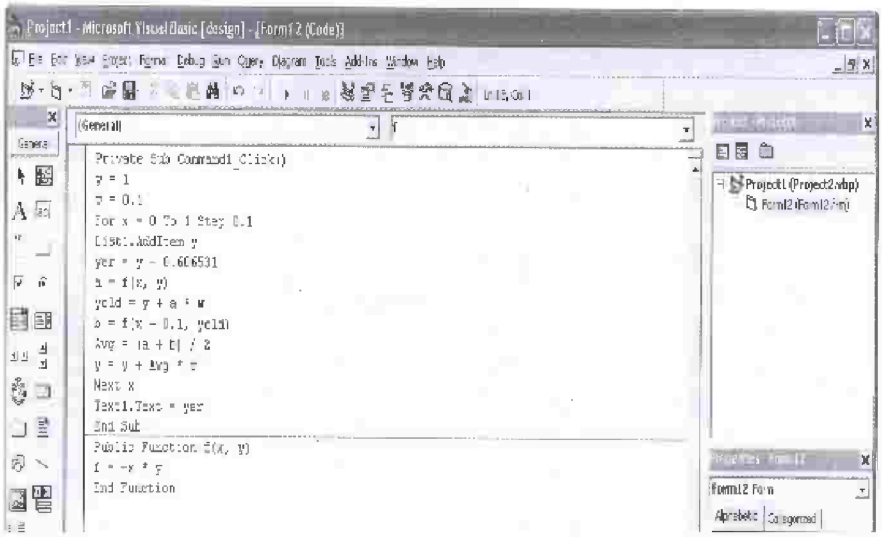
الشكل (12.8) مقارنة مع الحل التحليلي للمثال (7.8) [أويلر المعدلة]

نعطي أيضاً بالشكل (13.8) اعتماد قيمة $y(1)$ على ω، حيث نلاحظ أنه كلما صغرنا في ω كلما كانت القيمة أقرب إلى القيمة المتوقعة و هو أمر واضح.



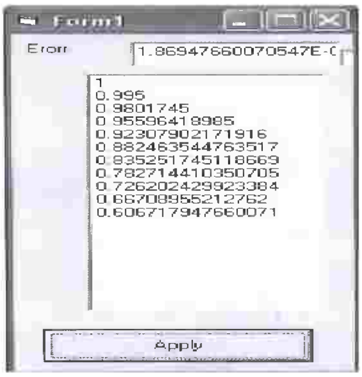
الشكل (13.8) اعتماد y على ω للمثال (7.8) [أويلر المعدلة]

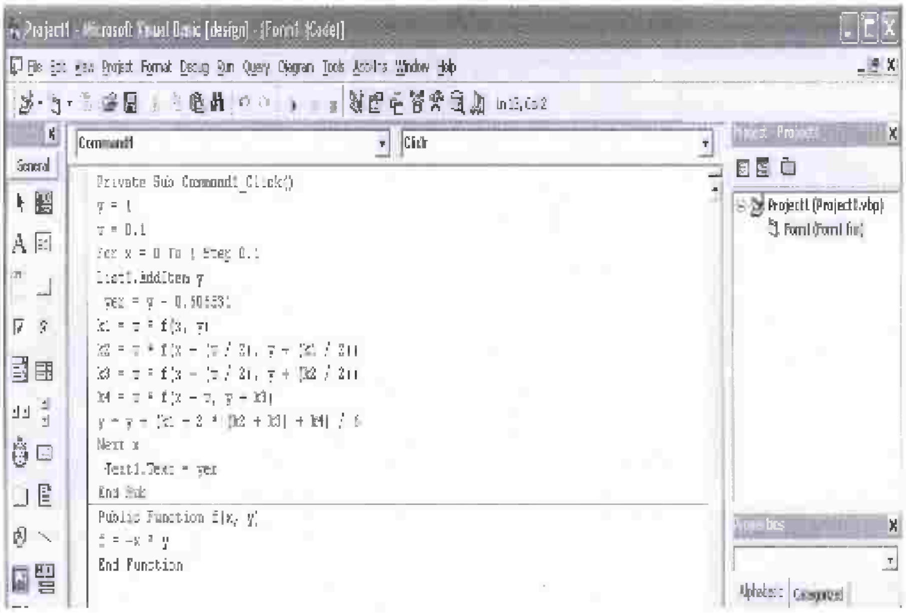
نوضح أيضا في الشكلين (14.8) و (15.8) و الجدولين (9.8) و (10.8) برامج ونتائج المثال (7.8) ولكن بطريقة بيسك المرئية وحيث نصل فيهم إلى نفس النتيجة المتوقعة.



الشكل (14.8) المثال (7.8) بلغة بيسك المرئية (أويلر المعدلة)

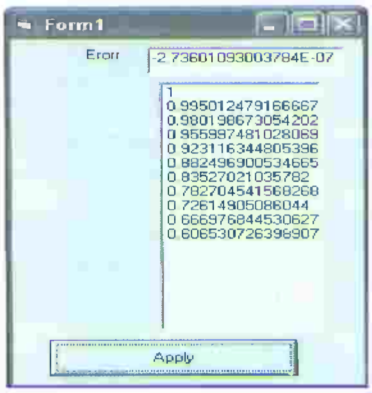
الجدول (9.8) نتائج المثال (7.8) بلغة بيسك المرئية (أويلر المعدلة)





الشكل (15.8) المثال (7.8) بلغة بيسك المرئية (رنج -كوتا)

الجدول (10.8) نتائج المثال (7.8) بلغة بيسك المرئية (رنج-كوتا)



8.8 معادلات تفاضلية من رتب عليا

إن الأسباب التي جعلتنا نعالج معادلات تفاضلية من الرتبة الأولى يمكن أن تتلخص فيما يلي:

أ) سهولة معالجة مثل هذه المعادلات واعتبارها الخطوة الأولى نحو معالجة بقية المسائل التي تحوي معادلات من الرتب العليا.

ب) شيوع مثل هذه المسائل (بمعادلات من رتبة أولى) في حقول شتى.

ج) إمكانية تحويل المعادلات من الرتب العليا إلى مجموعة من المعادلات من الرتبة الأولى كما سنوضحه فيما بعد.

في كثير من الحالات تواجهنا معادلات تفاضلية من النوع:

$$a_n \frac{d^n y}{dx^n} + a_{n-1} \frac{d^{n-1} y}{dx^{n-1}} + ... + a_1 \frac{dy}{dx} = \varphi(x, y) \qquad ...... (21.8)$$

وبحيث تعطي القيم الابتدائية للكميات $y, \frac{dy}{dx}, ..., \frac{d^{n-1} y}{dx^{n-1}}$ عند نقطة ما x_o.

أي أن المسألة التي لدينا تشمل معادلة تفاضلية عادية من الرتبة n. ولو استطعنا تحويل هذه المعادلة إلى مجموعة من المعادلات الآنية من الرتبة الأولى؛ فإنه يمكننا الاستفادة مما تمت دراسته حول الحلول العددية للمعادلات التفاضلية من الرتبة الأول وإيجاد حل المسألة تحت الدراسة.

وهذا ممكن، لو وضعنا المعادلة (21.8) على الصورة:

$$\frac{d^n y}{dx^n} = f\left(x, y, y^{(1)}, y^{(2)}, ..., y^{(n-1)}\right) \qquad ......(22.8)$$

ثم نقوم بالتحويلات التالية:

$$\begin{aligned} y_1 &= y \\ y_2 &= y_1^{(1)}\left(= y^{(1)}\right) \\ y_3 &= y_2^{(1)}\left(= y^{(2)}\right) \\ &\vdots \\ y_n &= y_{n-1}^{(1)}\left(= y^{(n-1)}\right) \end{aligned}$$

عندئذ نرى أن:

$$y_n^{(1)} = y^{(n)} = f\left(x, y_1, y_2, ..., y_n\right) \qquad ......(23.8)$$

وهكذا نرى أننا حصلنا على n من المعادلات الآنية من الرتبة الأولى و هي:

$$y_1^{(1)} = y_2 \ , \ y_2^{(1)} = y_3 \ , \ y_{n-1}^{(1)} = y_n$$

و

$$y_n^{(1)} = f\left(x, y_1, ..., y_n\right)$$

مثال (8.8)

لو كان لدينا المعادلة:

$$\frac{d^2 y}{dx^2} + \frac{dy}{dx} = y^2$$

فإنه يمكننا تحويل هذه المعادلة من الرتبة الثانية إلى معادلتين من الرتبة الأولى وذلك كما يلي:

نضع $y_1 = y$ و $y_2 = y_1^{(1)}$ ومنها نرى أن:

$$y_2^{(1)} = y_2^2 - y_1$$

وهكذا نوجد الحل عددياً إذا ما أعطينا قيم $y_1(= y)$ و $y_2\left(= \dfrac{dy}{dx}\right)$ عند نقطة ابتدائية ما.

وعموماً نستطيع تحويل أي معادلة من الرتبة n إلى n من المعادلات التفاضلية الآنية من الشكل :

$$\begin{aligned} y_1^{(1)} &= f_1(x, y_1, y_2, ..., y_n) \\ y_2^{(1)} &= f_2(x, y_1, y_2, ..., y_n) \\ &\vdots \\ y_n^{(1)} &= f_n(x, y_1, y_2, ..., y_n) \end{aligned}$$

وهي معادلات تفاضلية من الرتبة الأولى . بعدئذ نستطيع تطبيق الطرق المختلفة التي سبق وأن تمت دراستها بالبنود السابقة كطريقة رنج - كوتا، ونقوم بالعمليات التكرارية على كل معادلة.

ويمكن توضيح ذلك كما يلي:

لو كانت المعادلتان الآنيتان هما

$$\begin{aligned} y^{(1)} &= f_1(x, y, z) \\ z^{(1)} &= f_2(x, y, z) \end{aligned}$$

فإن العمليتين التكراريتين ، باستخدام طريقة RKM هما:

$$y_{i+1} = y_i + \frac{1}{6}\left[k_1 + 2k_2 + 2k_3 + k_4\right]$$

و

$$z_{i+1} = z_i + \frac{1}{6}\left[l_1 + 2l_2 + 2l_3 + l_4\right]$$

حيث:

$$\begin{aligned}
k_1 &= f_1(x_i, y_i, z_i)\omega \\
l_1 &= f_2(x_i, y_i, z_i)\omega \\
k_2 &= f_1\left(x_i + \frac{1}{2}\omega, y_i + \frac{1}{2}k_1, z_i + \frac{1}{2}l_1\right)\omega \\
l_2 &= f_2\left(x_i + \frac{1}{2}\omega, y_i + \frac{1}{2}k_1, z_i + \frac{1}{2}l_1\right)\omega \\
k_3 &= f_1\left(x_i + \frac{1}{2}\omega, y_i + \frac{1}{2}k_2, z_i + \frac{1}{2}l_2\right)\omega \\
l_3 &= f_2\left(x_i + \frac{1}{2}\omega, y_i + \frac{1}{2}k_2, z_i + \frac{1}{2}l_2\right)\omega \\
k_4 &= f_1(x_i + \omega, y_i + k_3, z_i + l_3)\omega \\
l_4 &= f_2(x_i + \omega, y_i + k_3, z_i + l_3)\omega
\end{aligned}$$

في ختام هذا البند نود أن نشير إلى أنه يمكن استخدام متسلسلة تايلور لإيجاد حلول المعادلات الآنية التي حصلنا عليها. نوضح استخدام هذه الطريقة بالمثال الآتي:

مثال (9.8)

أوجد الحل العددي للمعادلين $z^{(1)} = y, y^{(1)} = z$ عند $x = 0.1$ علماً بأن $z_o = 1, y_o = 0$ عند $x_o = 0$.

الحل:

من مفكوك تايلور (أو ماكلورين في هذه الحالة) نرى أن:

$$y = y_o + xy_o^{(1)} + \frac{x^2}{2!}y_o^{(2)} + \frac{x^3}{3!}y_o^{(3)} + ...$$

و

$$z = z_o + xz_o^{(1)} + \frac{x^2}{2!}z_o^{(2)} + \frac{x^3}{3!}z_o^{(3)} + ...$$

ولكي نوجد قيم z, y عند $x = 0.1$ علينا إيجاد قيم المشتقات المختلفة عند $x_o = 0$ ؛ولكن هذه يمكن حسابها كما يلي:

حيث أن:

$$z^{(1)} = y, y^{(1)} = z$$

فإنه، بالتعويض، نجد أن $y_o{}^{(1)} = z_0 = 1$ و $z_o{}^{(1)} = y_o = 0$.

نفاضل الآن المعادلتين الآنيتين لنحصل على:

$$z^{(2)} = y^{(1)} = z \quad \text{و} \quad y^{(2)} = z^{(1)} = y$$

ومنها نجد أن:

$$y_o{}^{(2)} = 0 \quad \text{و} \quad z_o{}^{(2)} = 1 \; .$$

نفاضل مرة أخرى لنحصل على:

$y^{(3)} = z$ و $z^{(3)} = y$ ومنها $y_o{}^{(3)} = 1$ و $z_o{}^{(3)} = 0$.

..... وهكذا.

وبالرجوع إلى مفكوك تايلور نستطيع حساب y و z عند $x = 0.1$ و حيث نرى أن:

$$y = 0 + (0.1)1 + \frac{(0.1)^2}{2!}(0) + \frac{(0.1)^3}{3!}(1) + ...$$

$$z = 1 + (0.1)(0) + \frac{(0.1)^2}{2!}(1) + \frac{(0.1)^3}{3!}(3) + ...$$

أو أن : $z \cong 1.005, y \cong 0.1002$

لاحظ أنه باستخدام هذه الطريقة لإيجاد الحل العددي كانت كل مشتقة معتمدة على قيم المشتقات السابقة وبذلك يجب الحذر في حساب المشتقات الدنيا حتى لا نقع في الخطأ الذي سينتشر إذ حدث.

9.8 طريقة الرمي (Shooting Method)

درسنا فيما سبق، وببعض التفصيل، حلول المعادلات التفاضلية العادية من الرتبة الأولى ولقد رأينا أنه لإيجاد الحل الوحيد يتعين علينا معرفة ثابت واحد (هو قيمة y عند نقطة ابتدائية ما). بالمثل لإيجاد حل المعادلات التفاضلية من الرتبة الثانية علينا معرفة ثابتين وهكذا . . .

في المعتاد تكون هذه الثوابت أو القيم معرفة كما يلي:

1. تكون الدالة أو مشتقتها معطاة عند نقطة البداية وتسمى المسألة بمسألة القيم الابتدائية مثل:

$$\frac{d^2x}{dt^2}+x=0 \quad ; \quad x(0)=0 \quad , \quad \left.\frac{dx}{dt}\right|_0=0$$

2. تكون الدالة أو مشتقتها معرفة عند نقطتين مختلفتين وعادة ما تكون هذه النقاط حدودية أي عند حدود نطاق المسألة؛ وفي هذه الحالة تسمى المسألة بمسألة القيم الحدية كالمسألة:

$$\frac{d^2y}{dx^2}+y=0 \qquad 0\le x\le 1$$

و

$$y(0)=0 \quad , \quad y(1)=1$$

والمسائل التي تدخل ضمن هذا النطاق عديدة وبميادين شتى كالفيزياء والهندسة وغيرها مثل تدفق الحرارة والحركة الاهتزازية و مسائل الجهد .

في هذا البند والبند الموالي ندرس الكيفية التي يتم بها استخدام بعض الطرق ،مثل طريقة الرمي وطريقة الحل من خلال مجموعة من المعادلات الآنية، لإيجاد حلول مسائل القيم الحدية بالمعادلات التفاضلية العادية.

لنأخذ المثال التالي:

مثال (10.8)

أوجد الحل العددي لمسألة القيم الحدية:

$$\frac{d^2y}{dx^2}-\left(1-x/5\right)y=x \quad ; \quad y(1)=2 \quad , \quad y(3)=-1$$

وذلك باستخدام طريقة الرمي.

الحل:

لو كان لدينا، بالإضافة قيمة $y^{(1)}(1)$ لاستطعنا اعتبار هذه المسألة مسألة قيم ابتدائية ولعالجناها بالطرق السابقة بعد تحويلها إلى معادلات آنية في y و $y^{(1)} = \frac{dy}{dx}$. ولكن ما هو معلوم لدينا هو قيم y عند $x = 1,3$.وعليه لو قمنا بعملية تخمين لقيمة $y^{(1)}$ عند $x = 1$ ثم قمنا بالحسابات، لنرى مدى تطابق القيمة المحسوبة لـ $y(3)$ ومع القيمة المعطاة،لتمكنا من الوصول للمنحنى المطلوب بعد بعض عناء.

دعنا نأخذ $y^{(1)}(1) = -1.5$ ؛عندئذ لو قمنا بحل المسألة كمسألة قيم ابتدائية وغضضنا الطرف عن $y(3)$ لحصلنا على $y(3) = 4.811$.(لاحظ أننا استعملنا $\Delta x = 0.2$) وهي قيمة عالية مقارنة بالقيمة المعطاة $y(3) = -1$. نحاول بقيمة أخرى أقل لـ $y^{(1)}$ مثل $y^{(1)} = -3$. إذا فعلنا ذلك حصلنا على $y(3) = 0.453$ ، وهي أيضاً قيمة كبيرة نسبيا ولكنها معقولة.

الآن ولتخمين القيمة السليمة والتي توصلنا للحل نستعمل الاستكمال الخطي، من خلال القيمتين السابقتين، لنحصل على $y(3) = -1$ بالضبط؛ كما نحصل على منحنى الحل الموضح بالجدول (11.8).

الجدول(11.8) الحل العددي للمثال (10.8) باستخدام طريقة الرمي

x	1	1.2	1.4	1.6	1.8	2.0	2.2	2.4	2.6	2.8	3.0
y	2.0	1.348	0.787	0.305	-0.104	-0.443	-0.712	-0.908	-1.026	-1.06	-1.0

من خلال هذا المثال يتضح السبب في تسمية الطريقة بطريقة الرمي؛ كما يمكن تبين العلاقة الوطيدة بين ما تم استخدامه في هذه الطريقة وبين ما يستخدم عند رمي القذائف المدفعية. وعموما يمكننا تلخيص الطريقة في الخطوات التالية:

1. لحل مسألة قيم حدية حاول أن تكون مسألة قيم ابتدائية بافتراض شروط كافية.
2. أوجد حل هذه المعادلة الجديدة وقارن القيمة المحسوبة مع الشروط عند الحدود الأخرى.
3. أعد تغيير هذه القيم الابتدائية حتى تحصل على تطابق مع الشروط الحدية كلها.

مثال (11.8)

مستعملاً طريقة الرمي أوجد حل مسألة القيم الحدية:

$$\frac{d^2y}{dx^2} = y \quad ; \quad y(1) = 1.1752 \quad y(3) = 10.0179$$

الحل:

بناء على ما تم التقديم له نكتب البرنامج الموضح أسفله بالشكل (16.8) والذي يستعمل $x1$ لـ y و $x2$ لـ $\frac{dy}{dx}(x = T)$ كما يستخدم البرنامج الفرعي SUBROUTINE RKSYST (DERIVS,TO,H,XO,XEND,XWRK,F,N) والذي يحسب حلول مجموعة من المعادلات الآنية من الرتبة الأولى بطريقة رنج-كوتا؛ وهذا بدوره يستعمل البرنامج الفرعي SUBROUTINE DERIVS (X,T,F,N) الذي يحسب المشتقات.

الجدول (12.8) يوضح النتائج المتحصل عليها بإجراء البرنامج المذكور، ومنه نرى

مدى الدقة التي حصلنا عليها باستخدام طريقة الرمي، حيث نلاحظ أن الأخطاء صغيرة جداً عندما نقارن الحل بالحل التحليلي $y = \sinh x$.

```
      DIMENSION XO (2),XEND(2),XWRK(4,2),F(2)
      EXTERNAL DERIVS

      DATA H,N,G1,G2,TOL/ 1,2,1.5431,1.,.001/

      DATA TSTART,XSTART,D/1.,1.1752,10.0179/

      XO(1)= XSTART
      TO   =TSTART
      XO(2)=G1

      DO 10 I=1,20
      CALL RKSYST (DERIVS,TO,H,XO,XEND,XWRK,F,N)

      XO(1)=XEND(1)
      XO(2)=XEND(2)
      TO=TO+H

   10 CONTINUE

      R1=XO(1)
      TO=TSTART
      XO(1)=XSTART
      XO(2)=G2

      DO 20 I=1,20
      CALL RKSYST(DERIVS,TO,H,XO,XEND,XWRK,F,N)

      XO(1)=XEND(1)
      XO(2)=XEND(2)
      TO=TO+H

   20 CONTINUE

      R2=XO(1)

      DO 40 ITER=1,20
      IF (ABS(R2-D).LE.TOL)GOTO 99

      TO   = TSTART
      XO(1)= XSTART

      XO(2)=G1+(G2-G1)/(R2-R1)*(D-R1)
      G1=G2
      G2=XO(2)
      R1=R2

      DO 30 I=1,20
      CALL RKSYST(DERIVS,TO,H,XO,XEND,XWRK,F,N)

      XO(1)=XEND(1)
      XO(2)=XEND(2)
      TO=TO+H

   30 CONTINUE
      R2=XO(1)
   40 CONTINUE

      GOTO 100
```

```
100 WRITE(*,202)H2,G2

202 FORMAT(1HO'FINAL VALUE AT END OF INTERVAL',2X,F7.4,//,1HO'USING INITIAL SLOP VALUE OF',2
X,F7.4,2X,'LAST COMPUTATIONS WERE',//)


    TO=TSTART
    XO(1)=XSTART
    XO(2)=G2


    WRITE(*,203)TO,XO(1),XO(2),SINH(TO),ERROR
203 FORMAT (1HO,10X,'TIME',5X,'X1 VALUE',5X,'X2 VALUE',5X,'SINH(TIME)',4X,'ERROR',/,1H,3X,F7
.4,6X,F7.4,6X,F7.4,6X,F7.4,6X,F7.4)


    DO 50 I=1,20
    CALL RKSYST(DERIVS,TO,H,XO,XEND,XWRK,F,N)

    XO(1)=XEND(1)
    XO(2)=XEND(2)
    TO=TO+H
    S=SINH(TO)
    ERROR=S-XO(1)

    WRITE(*,204)TO,XO(1),XO(2),SINH (TO),ERROR
204 FORMAT (1H,5X,F7.4,6X,F7.4,6X,F7.4,6X,F7.4,6X,F7.4)
50  CONTINUE
    STOP
    END

    SUBROUTINE RKSYST(DERIVS,TO,H,XO,XEND,XWRK,F,N)

    DIMENSION XO(N),XEND(N),XWRK(4,N),F(N)
    CALL DERIVS(XO,TO,F,N)
    DO 10 I=1,N
    XWRK(1,I)=H*F(I)
    XEND(I)=XO(I)+XWRK(1,I)/2.
10 CONTINUE

    CALL DERIVS(XEND,TO+H/2.,F,N)
    DO 20 I=1,N
    XWRK(2,I)=H*F(I)
    XEND(I)=XO(I)+XWRK(2,I)/2
20  CONTINUE
    CALL DERIVS(XEND,TO+H/2.,F,N)
    DO 30 I=1,N
    XWRK(3,I)=H*F(I)
    XEND(I)=XO(I)+XWRK(3,I)
30  CONTINUE
    CALL DERIVS(XEND,TO+H,F,N)
    DO 40 I=1,N
    XWRK(4,I)=H*F(I)
40 CONTINUE

    DO 50 I=1,N


  XEND(I)=XO(I)+(XWRK(1,I)+2.*XWRK(2,I)+2.*XWRK(3,I)+XWRK(4,I))/6

50  CONTINUE
    RETURN
    END


    SUBROUTINE DERIVS(X,TO,F,N)
    DIMENSION X(N),F(N)
    F(1)=X(2)
    F(2)=X(1)
    RETURN
    END
```

الشكل (16.8) حل المثال (11.8) باستخدام طريقة الرمي.

الجدول (12.8) نتائج المثال (11.8) باستخدام طريقة الرمي.

TOLERENCE CRITERION MET IN 2 ITERATIONS
FINAL VALUE AT END OF INTERVAL 10.0179

USING INITIAL SLOP VALUE OF 1.5431 LAST COMPUTATIONS WERE

TIME	X1 VALUE	X2 VALUE	SINH(TIME)	ERROR
1.0000	1.1752	1.5431	1.1752	.0000
1.1000	1.3356	1.6685	1.3356	.0000
1.2000	1.5095	1.8107	1.5095	.0000
1.3000	1.6984	1.9709	1.6984	.0000
1.4000	1.9043	2.1509	1.9043	.0000
1.5000	2.1293	2.3524	2.1293	.0000
1.6000	2.3756	2.5775	2.3756	.0000
1.7000	2.6456	2.8283	2.6456	.0000
1.8000	2.9422	3.1075	2.9422	.0000
1.9000	3.2682	3.4177	3.2682	.0000
2.0000	3.6269	3.7622	3.6269	.0000
2.1000	4.0219	4.1443	4.0219	.0000
2.2000	4.4571	4.5679	4.4571	.0000
2.3000	4.9370	5.0372	4.9370	.0000
2.4000	5.4662	5.5570	5.4662	.0000
2.5000	6.0502	6.1323	6.0502	.0000
2.6000	6.6947	6.7690	6.6947	.0000
2.7000	7.4063	7.4735	7.4063	.0000
2.8000	8.1919	8.2528	8.1919	.0000
2.9000	9.0596	9.1146	9.0596	.0000
3.0000	10.0179	10.0677	10.0179	.0000

Stop - Program terminated

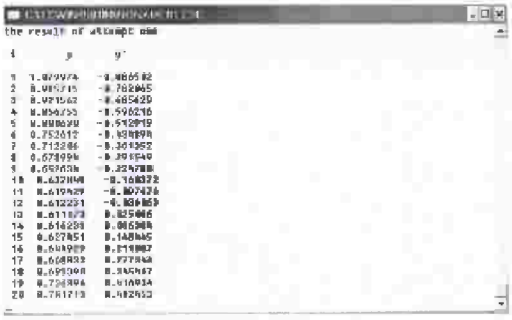
في الشكل(17.8) نعطي برنامجا حاسوبيا بلغة C لنفس المثال (11.8)

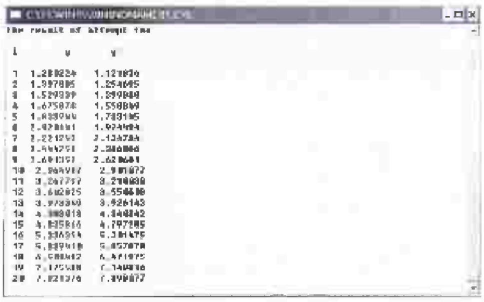
```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void rksyst();
void derivs(float x[]);
float xo[2],xend[2],xwrk[4][2],f[2], h;
int n ;
double to;

void main(){
double s;
float g1,g2,tstart,xstart,d,r1,r2,error;
int i,j;
clrscr();
h=0.1;  n=2; g1=-1;  g2=1;
tstart=1.0;  xstart=1.1752;  d=10.0179;
//************************
 //* WE SUPPOS  TO FIND Y *//
//************************
xo[0]=xstart;
to=tstart;
xo[1]=g1;
printf("\nthe result of attempt one\n\n");
printf(" i          y          y'\n\n");
for(i=1;i<=20;i++)
{
rksyst();
xo[0]=xend[0];
xo[1]=xend[1];
to=to+h;
printf(" %d  %f   %f\n",i,xo[0],xo[1]);
}
getch();
clrscr();

//*************************************
 //* WE SUPPOS ANATHER VALUE TO FIND Y *//
//*************************************
r1=xo[0];
to=tstart;
xo[0]=xstart;
xo[1]=g2 ;
printf("\nthe result of attempt two\n\n");
printf(" i          y          y'\n\n");
for(i=1;i<=20;i++)
{
rksyst();
xo[0]=xend[0];
xo[1]=xend[1];
to=to+h;
printf(" %d  %f   %f\n",i,xo[0],xo[1]);
}
getch();
clrscr();
//*******************
 //* LINEAR CREATION *//
//*******************
r2=xo[0];
 printf("\nthe result of LINEAR CREATION\n\n");
printf(" i          y          y'\n\n");
for(j=1;j<=20;j++)
{
 if(fabs(r2-d)<=0.001) goto  one;
 to=tstart;
 xo[0]=xstart;
 xo[1]=g1+(g2-g1)/(r2-r1)*(d-r1);
 g1=g2;
 g2=xo[1];
```

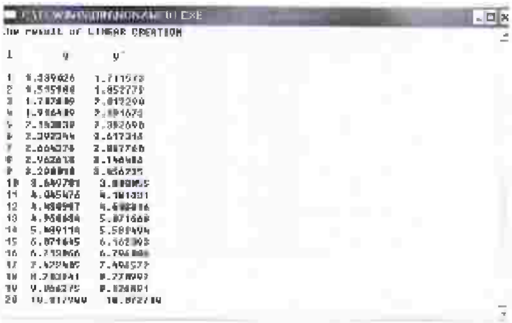
```
  r1=r2;
   for(i=1;i<=20;i++)
          {
          rksyst();
          xo[0]=xend[0];
          xo[1]=xend[1];
          to=to+h;
 printf(" %d  %f   %f\n",i,xo[0],xo[1]);
 }
 getch();
 clrscr();
 r2=xo[0];
 }
 one:
 to=tstart;
 xo[0]=xstart;
 xo[1]=g2;
 printf("\n the result of shooting method comper with sinch function \n\n");
 printf(" x        y          y'       s       error\n\n");
   for(i=1;i<=20;i++)
          {
          rksyst();
          xo[0]=xend[0];
          xo[1]=xend[1];
          to=to+h;
          s=sinh(to);
          error=s-xo[0];
          printf(" %f   %f   %f   %f   %f\n",to,xo[0],xo[1],s,error);
          }
 getch();
 }
 //******************
  //* RK APPLICATION *//
 //******************
 void rksyst()
 {  int i;
   derivs(xo);
   for(i=0;i<n;i++)
   {
    xwrk[0][i]=h*f[i];
    xend[i]=xo[i]+xwrk[0][i]/2;
   }
   derivs(xend);
   for(i=0;i<n;i++)
   {
    xwrk[1][i]=h*f[i];
    xend[i]=xo[i]+xwrk[1][i]/2;
   }
   derivs(xend);
   for(i=0;i<n;i++)
   {
    xwrk[2][i]=h*f[i];
    xend[i]=xo[i]+xwrk[2][i]/2;
   }
    derivs(xend);
   for(i=0;i<n;i++)
   {
    xwrk[3][i]=h*f[i];
     }
  for(i=0;i<n;i++)
    xend[i]=xo[i]+(xwrk[0][i]+2*xwrk[1][i]+2*xwrk[2][i]+xwrk[3][i])/6;
  }
  //******************
 void derivs(float x[])
 {
   f[0]=x[1];
   f[1]=x[0];
 }
```

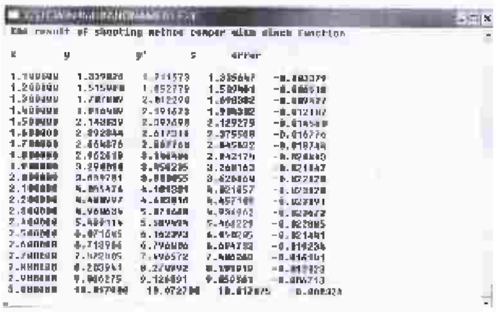
الشكل(17.8)- المثال(11.8) بلغة C

الجدول (**13.8**) نتائج المثال (**11.8**) باستخدام طريقة الرمي بلغة **C**.









كما نعطي في الشكل(**18.8**) برنامجا حاسوبيا بلغة بيئة التطوير(دلفي) لنفس المثال(**11.8**).و نتائج تنفيذ هذا البرنامج معطاة بالجدول (**14.8**)

```
procedure Shooting_Method;
var
  XXO, XEnd, F : TTwo;
  XWrk : TFourTwo;
  Iter, i, N : integer;
  R1,R2,RErro,S,H,TTO,G1,G2,Tol,TStart,XStart,D : real;
  Label L100;
  Label L99;
begin
  H := 0.1; N := 2; G1 := -1.0; G2 := 1.0; Tol := 0.001;
  TStart := 1.0; XStart := 1.1752; D := 10.0179;
  XXO[1] := XStart;
  TTO := TStart;
  XXO[2] := G1;

  for i := 1 to 20 do
  begin
    RKSyst(TTO, H, XXO, XEnd,  F, XWrk, N);

    XXO[1] := XEnd[1];
    XXO[2] := XEnd[2];
    TTO := TTO + H;
  end;

    R1 := XXO[1];
    TTO := TStart;
    XXO[1] := XStart;
    XXO[2] := G2;

  for i := 1 to 20 do
  begin


      RKSyst(TTO, H, XXO, XEnd,  F, XWrk, N);
      XXO[1] := XEnd[1];
      XXO[2] := XEnd[2];
      TTO := TTO + H;
    end;

    R2 := XXO[1];

    for Iter := 1 to 20 do
    begin
   //...........................
      if Abs(R2-D) < Tol then goto L99;
      TTO := TStart;
      XXO[1] := XStart;
      XXO[2] := G1 + (G2-G1) / (R2-R1) * (D-R1);
      G1 := G2;
      G2 := XXO[2];
      R1 := R2;
   //...........................
      for i := 1 to 20 do
      begin
```

```
        RKSyst(TTO, H, XXO, XEnd,  F, XWrk, N);
        XXO[1] := XEnd[1];
        XXO[2] := XEnd[2];
        TTO := TTO + H;
      end;  //for i
      R2 := XXO[1];
    end;  //for Iter

    memo1.Lines.Add('We did not meet tolerance criteriion in
  20 iterations. Final value at end interval was');

    goto L100;

    L99:
    memo1.Lines.add('Tolerence criterion met in ' +
  inttostr(Iter) + ' iterations ');
    L100:
    memo1.Lines.add('Final Value at end of interval was ' +
  floattostr(R2) );
    memo1.Lines.add('Using initial slope value of ' +
  floattostr(G2) + ' last computaions ' );

    TTO := TStart;
    XXO[1] := XStart;
    XXO[2] := G2;

    S := Sinh(TTO);
    RErro := S - XXO[1];


  memo1.Lines.add(floattostr(TTO) + '  ' +
floattostr(XXO[1]) + '   ' + floattostr(XXO[2]) + '   ' +
floattostr(S) + '  ' + floattostr(RErro) );
  for i := 1 to 20 do
  begin
      RKSyst(TTO, H, XXO, XEnd,  F, XWrk, N);
      XXO[1] := XEnd[1];
      XXO[2] := XEnd[2];
      TTO := TTO + H;
      S := Sinh(TTO);
      RErro := S - XXO[1];
      memo1.Lines.add(floattostr(TTO) +  '   ' +
floattostr(XXO[1]) + '     ' + floattostr(XXO[2]) + '
' + floattostr(S) + '       ' + floattostr(RErro) );
  end;


end;

procedure RKSyst(TTO, H : real; var XXO, XEnd, F : TTwo;
var XWrk : TFourTwo; N : integer);
var
  i, j : integer;
begin
```

```
    Derivs(XXO, TTO, F , N);
    for i := 1 to N do
    begin
      XWrk[1,i] := H * F[i];
      XEnd[i] := XXO[i] + XWrk[1,i] / 2;
    end;

    Derivs(XEnd, TTO + H/2, F , N);
    for i := 1 to N do
    begin
      XWrk[2,i] := H * F[i];
      XEnd[i] := XXO[i] + XWrk[2,i] / 2;
    end;

    Derivs(XEnd, TTO + H/2, F , N);
    for i := 1 to N do
    begin
      XWrk[3,i] := H * F[i];
      XEnd[i] := XXO[i] + XWrk[3,i] / 2;
    end;

    Derivs(XEnd, TTO + H, F , N);
    for i := 1 to N do
    begin
      XWrk[4,i] := H * F[i];
    end;

  for i := 1 to N do
  begin
    XEnd[i] := XXO[i] + (XWrk[1,i] + 2.0 * XWrk[2,i] + 2.0
* XWrk[3,i] + XWrk[4,i])/6
  end;

end;

procedure Derivs(var X : TTwo; T : real; Var F : TTwo;   N
: integer);
begin
  F[1] := X[2];
  F[2] := X[1];
end;
```

الشكل(**18.8**) – المثال(**11.8**) بلغة دلفي

الجدول (**14.8**) نتائج المثال (**11.8**) باستخدام طريقة الرمي بلغة دلفي.

Tolerance criterion met in 2 iterations

Final Value at end of interval was 10.0179

Using initial slope value of 1.58728941708334 last computations

Time	X1 Value	X2 Value	Sinh(Time)	RError
1.0	1.1752	1.58728941708334	1.1752011936438	1.193643801356E-6
1.1	1.33902646786547	1.7115733298408	1.33564747012418	-0.00337899776129236
1.2	1.51597981412378	1.85277847625661	1.50946135541217	-0.0065184587116045
1.3	1.7078069998091	2.01228972541865	1.69838243729262	-0.00942656261648267
1.4	1.91640893743501	2.19167283432739	1.90430150145153	-0.0121074359834776
1.5	2.14383920972997	2.39269981531617	2.12927945509482	-0.0145597546351555
1.6	2.39234440719407	2.61731624185798	2.37556795320023	-0.0167764539938364
1.7	2.66437628165759	2.86776066271153	2.64563193383723	-0.0187443478203537
1.8	2.96261799663574	3.14648631517518	2.94217428809568	-0.0204436485400589
1.9	3.29001029246601	3.45523535090722	3.26816291152632	-0.0218473809376886
2.0	3.64978108738231	3.80005581255947	3.62686040784702	-0.0229206795352863
2.1	4.04547870052285	4.18133162660223	4.02185674215734	-0.0236198583655123
2.2	4.4809971106317	4.60381590746048	4.4571051705359	-0.0238919400659046
2.3	4.96063433507113	5.07166790072671	4.93696180554596	-0.0236725295251654
2.4	5.48911472801572	5.58949392908886	5.4662292136761	-0.0228855143396167
2.5	6.07164555468796	6.16239274404531	6.0502044810398	-0.0214410736491607
2.6	6.7139663005917	6.79600572996119	6.69473222839389	-0.0192340721980102
2.7	7.42240522131985	7.49657245469471	7.40626310806855	-0.0161421152530998
2.8	8.20394169011234	8.27099211389985	8.19191835423593	-0.0120233358764086
2.9	9.06627495743722	9.12689147399716	9.05956107469334	-0.00671388274387397
3.0	10.0179	10.0726999829428	10.0178749274099	-2.50725900770021E-5

أخيرا نعطي برنامجا حاسوبيا لنفس المثال بلغة بيسك و ذلك بالشكل(**19.8**)؛أما النتائج فهي معطاة بالجدول(**15.8**).

```
DECLARE FUNCTION R! (X!)
DECLARE FUNCTION Q! (X!)
DECLARE FUNCTION P! (X!)
DIM Y1(200), Y2(200), Y(200), ER(200)

CLS
SCREEN 12
OPEN "GAM.TXT" FOR OUTPUT AS #2
W = .1
B = 3
A = 1
N = 20
BETA = 10.0179
Y1(1) = 1.1752
Y11 = 0
Y2(1) = 0
Y22 = 1
X = A
FOR I = 2 TO N + 1
K11 = W * Y11
K12 = W * (P(X) * Y11 + Q(X) * Y1(I - 1) + R(X))
K13 = W * Y22
K14 = W * (P(X) * Y22 + Q(X) * Y2(I - 1))

K21 = W * (Y11 + .5 * K12)
K22 = W * (P(X + .5 * W) * (Y11 + .5 * K12) + Q(X + .5 * W) * (Y1(I - 1) + .5 *
K11) + R(X + .5 * W))
K23 = W * (Y22 + .5 * K14)
K24 = W * (P(X + .5 * W) * (Y22 + .5 * K14) + Q(X + .5 * W) * (Y2(I - 1) + .5 *
K13))

K31 = W * (Y11 + .5 * K22)
K32 = W * (P(X + .5 * W) * (Y11 + .5 * K22) + Q(X + .5 * W) * (Y1(I - 1) + .5 *
K21) + R(X + .5 * W))
K33 = W * (Y22 + .5 * K24)
K34 = W * (P(X + .5 * W) * (Y22 + .5 * K24) + Q(X + .5 * W) * (Y2(I - 1) + .5 *
K23))
```

```
K41 = W * (Y11 + K32)
K42 = W * (P(X + W) * (Y11 + K32) + Q(X + W) * (Y1(I - 1) + K31) + R(X + W))
K43 = W * (Y22 + K34)
K44 = W * (P(X + W) * (Y22 + K34) + Q(X + W) * (Y2(I - 1) + K33))


Y1(I) = Y1(I - 1) + (K11 + 2 * K21 + 2 * K31 + K41) / 6
Y11 = Y11 + (K12 + 2 * K22 + 2 * K32 + K42) / 6
Y2(I) = Y2(I - 1) + (K13 + 2 * K23 + 2 * K33 + K43) / 6
Y22 = Y22 + (K14 + 2 * K24 + 2 * K34 + K44) / 6


X = A + (I - 1) * W
NEXT I

C = (BETA - Y1(N + 1)) / Y2(N + 1)
X = A
FOR I = 2 TO N + 1
X = A + (I - 1) * W
Y(I) = Y1(I) + C * Y2(I)
NEXT I
X = A

C = (BETA - Y1(N + 1)) / Y2(N + 1)
Y(I) = Y1(1) + C * Y2(1)
YEX = 1 / 2 * (EXP(X) - EXP(-X))
ER(I) = ABS((Y(I) - YEX) / YEX) * 100
PRINT #2, "    X         Y(I)           YEXACT      ERROR"
PRINT #2, "------      -------       -------      ---------"
PRINT ,
PRINT #2, USING "##.##    ##.#########    ##.#########  #.######^^^^"; X; Y(I);
YEX; ER(I)

FOR I = 2 TO N + 1


X = A + (I - 1) * W
YEX = 1 / 2 * (EXP(X) - EXP(-X))
ER(I) = ABS((Y(I) - YEX) / YEX) * 100
PRINT #2, USING "##.##    ##.#########    ##.#########  #.######^^^^"; X; Y(I);
YEX; ER(I)
PRINT USING "##.##    ##.#########    ##.#########  #.######^^^^"; X; Y(I); YEX;
ER(I)
NEXT I
END

FUNCTION P (X)
P = 0
END FUNCTION

FUNCTION Q (X)
Q = 1
END FUNCTION

FUNCTION R (X)
R = 0
END FUNCTION
```

الشكل(19.8) – المثال(11.8) بلغة بيسك

الجدول (15.8) نتائج المثال (11.8) باستخدام طريقة الرمي بلغة بيسك.

X	Y(I)	YEXACT	ERROR
1.00	1.175199986	1.175201178	0.101437E-03
1.10	1.335647464	1.335647464	0.0000000E+00
1.20	1.509462357	1.509461403	0.6317977E-04
1.30	1.698384523	1.698382378	0.1263418E-03
1.40	1.904304624	1.904301405	0.1690200E-03
1.50	2.129283667	2.129279375	0.2015487E-03
1.60	2.375573158	2.375567913	0.2207981E-03
1.70	2.645638227	2.645632029	0.2343063E-03
1.80	2.942181826	2.942174196	0.2593115E-03
1.90	3.268171549	3.268162727	0.2699219E-03
2.00	3.626870155	3.626860380	0.2695213E-03
2.10	4.021867752	4.021856308	0.2845475E-03
2.20	4.457117558	4.457105160	0.2781574E-03
2.30	4.936975479	4.936961651	0.2800969E-03
2.40	5.466244698	5.466229916	0.2704232E-03
2.50	6.050221443	6.050204277	0.2837282E-03
2.60	6.694750786	6.694731712	0.2849029E-03
2.70	7.406282902	7.406263351	0.2639701E-03
2.80	8.191940308	8.191918373	0.2677579E-03
2.90	9.059584618	9.059561729	0.2526412E-03
3.00	10.017900467	10.017874718	0.2570326E-03

10.8 طريقة الفروق المحدودة (Finite Difference Method)

كما نعلم، يمكن كتابة مشتقة دالة بدلالة الفروق المحدودة وبذلك يمكننا التعويض عن المشتقة بهذه الفروق ونحصل على معادلة فرقية **Difference Equation** بدلاً من معادلة تفاضلية؛ وحلول هذه المعادلة الفرقية هي الحلول التقريبية للمعادلة المطلوبة.

وهذه الطريقة أفضل من الطريقة السابقة في كثير من الأحيان. لنعتبر نفس المثال السابق (10.8) ولنأخذ نفس المسألة :

$$\frac{d^2 y}{dx^2} - \left(1 - \frac{x}{5}\right) y = x \quad ; \quad y(1) = 2 \quad , \quad y(3) = -1$$

ولنعمل بالفروق المركزية حيث إنها أدق من الأمامية والخلفية وعليه نعوض عن $\frac{dy}{dx}$ و $\frac{d^2 y}{dx^2}$ بالمعادلتين:

$$\left.\frac{dy}{dx}\right|_{x=x_i} = \frac{y_{i+1} - y_{i-1}}{2h} + O\left(h^2\right)$$

$$\left.\frac{d^2 y}{dx^2}\right|_{x=x_i} = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + O\left(h^2\right)$$

و h هي قيمة الخطوة الثابتة بين قيم x . لو عوضنا في المعادلة المعطاة لحصلنا على:

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} - \left(1 - \frac{x_i}{5}\right) y_i = x_i$$

أو أن:

$$y_{i-1} - \left[2 + h^2\left(1 - \frac{x_i}{5}\right)\right] y_i + y_{i+1} = h^2 x_i$$

وحيث قمنا بوضع $y = y_i$ و $x = x_i$. وهكذا فنحن الآن أمام مشكلة حل هذه المعادلة الفرقية في الفترة $x \in [1,3]$.

نقسم الفترة إلى عدد من الفترات الفرعية، مثلاً نأخذ $h = \Delta x = 0.5$ فتكون بذلك قيم x هي:

$$x_1 = 1, x_2 = 1.5, x_3 = 2, x_4 = 2.5, x_5 = 3$$

ونكتب كل معادلة مناظرة لـ x_i على حدة وذلك كما يلي:

لـ $x = x_2 = 1.5$

$$y_1 - \left[2 + \frac{1}{2}\left(\frac{1}{2}\right)\left(1 - \frac{1.5}{5}\right)\right] y_2 + y_3 = \frac{1}{2}\left(\frac{1}{2}\right)(1.5)$$

لـ $x = x_3 = 2$

$$y_2 - \left[2 + \frac{1}{2}\left(\frac{1}{2}\right)\left(1 - \frac{2.0}{5}\right)\right]y_3 + y_4 = \left(\frac{1}{2}\right)\cdot\left(\frac{1}{2}\right)(2)$$

لـ $x = x_4 = 2.5$

$$y_3 - \left[2 + \frac{1}{2}\left(\frac{1}{2}\right)\left(1 - \frac{2.5}{5}\right)\right]y_4 + y_5 = \left(\frac{1}{2}\right)\cdot\left(\frac{1}{2}\right)(2.5)$$

وحيث أن $y_1 = 2$ و $y_5 = -1$ فإنه بالتعويض نحصل على:

$$\begin{pmatrix} -2.175 & 1 & 0 \\ 1 & -2.150 & 1 \\ 0 & 1 & -2.125 \end{pmatrix}\begin{pmatrix} y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} -1.625 \\ 0.5 \\ 1.625 \end{pmatrix}$$

وبحل هذه المعادلات نجد أن: $y_2 = 0.552\ ,\ y_3 = -0.424\ ,\ y_4 = -0.964$

ولو قارنا بحلول الطريقة الأولى (الرمي) فإننا نلاحظ أنه بالرغم من بساطة التجزئة (خمس نقاط) إلا أن الحلول كانت معقولة. وهذا يعني بالطبع أن اختيار قيمة h له دور كبير في الحصول على تقريب دقيق. وعليه لو اخترنا $h = 0.2$ وقمنا بإجراء نفس الخطوات المذكورة أعلاه وحسبنا القيم المختلفة لـ y فإننا نحصل على النتائج الموضحة بالجدول (13.8) والذي يوضح أيضاً مقارنة بين هذه الطريقة وطريقة الرمي.

وهكذا نلاحظ أن اختيار $h = \Delta x = 0.2$ أضفى دقة أكثر. نلاحظ أيضاً أن مقدار الخطأ يتناسب مع h^2 وذلك من خلال الطريقة المستخدمة.

الجدول(16.8) قيم y (المثال (10.8)) باستخدام طريقة الفروق المحدودة وطريقة الرمي

x	y (الفروق المحدودة)	y (الرمي)
1.0	2.0	2.0
1.2	1.351	1.348
1.4	0.792	0.787
1.6	0.311	0.305
1.8	-0.097	-0.104
2.0	-0.436	-0.443
2.2	-0.705	-0.712
2.4	-0.903	-0.908
2.6	-1.022	-1.026
2.8	-1.058	-1.060
3.0	-1.000	-1.000

في ختام هذا البند نلخص خطوات طريقة الفروق المحدودة فيما يلي:

1. لإيجاد حل مسألة قيم حدية، نستبدل المعادلة بمعادلة فرقية وذلك بالتعويض عن المشتقات بدلالة الفروق المركزية.
2. نجزئ الفترة إلى فترات فرعية مناسبة متساوية ونكتب المعادلة الفرقية عند كل نقطة حيث الدالة غير معروفة.
3. نوجد حل المعادلات الآنية الناتجة.

تمارين (8)

1. قارن بين طريقة أويلر وامتدادها والأكثر امتداداً.

2. ما هي الفروق الأساسية بين طريقة أويلر وطريقة (ملن) وطريقة رنج-كوتا؟

3. ما هي الطرق المستعملة في حل مسائل القيم الابتدائية وفي حل مسائل القيم الحدية؟

4. باستخدام طريقة أويلر وأويلر المعدلة وطريقة رنج -كوتا أوجد حل المسائل التالية:

أ) $\frac{dy}{dx} = \cos x \quad F(0) = 0, \quad F\left(\pi/2\right) = ?$.

ب) $\frac{dy}{dx} = 3x^2 \quad F(-1) = -1, \quad F(1) = ?$.

قارن أجوبة الطرق المختلفة و ناقش.

5. أوجد حل مسألة القيم الابتدائية:

$$\frac{dy}{dx} = -xy; \quad x_o = 0, \quad y_o = 1 .$$

وذلك عند $\bar{x} = 2$. خذ $\omega = 0.1$ (قارن بقيم $e^{-x^2/2}$)

6. في دائرة كهربية إذا كانت العلاقة بين الجهد V والتيار I والزمن t معطاة بالمعادلتين:

$$\frac{dI}{dt} = \frac{1}{L}V \quad , \quad \frac{dV}{dt} = -\frac{1}{c}I$$

حيث c, L ثوابت تخص الدائرة المعنية؛ وإذا كان $I(0) = I_o$ و $V(0) = V_o$.

فأوصف كيفية استخدام الطرق العددية المختلفة التي تم التطرق إليها في هذا الفصل لإيجاد I, V عند أي لحظة زمنية $t > 0$.

7. من قانون نيوتن للتبريد يكون معدل تبريد مادة بهواء متحرك يساوي نصف الفرق بين درجة حرارة المادة وتلك للهواء. و إذا كانت درجة حرارة الهواء هي $300^o K$ وأنه عند اللحظة $t = 0$ كانت درجة حرارة المادة $370^o K$ فأحسب متى تكون درجة الحرارة مساوية لـ $310^o K$.

8. إذا علم بأن معدل الزيادة في سكان بلد ما يتناسب مع عدد السكان (وحيث ثابت التناسب k يساوي 0.01) وأن عدد السكان تضاعف في 50 سنة. فأحسب متى يصبح عدد السكان ثلاثة أمثال السكان الأصليين.

9. احسب قيمة y ، عند $x = 4$ ، التي تحقق مسألة القيم الابتدائية التالية:

$$\frac{d^2 y}{dx^2} = y \quad , \quad \left.\frac{dy}{dx}\right|_o = -1 \quad , \quad y(0) = 1$$

(قارن إجابتك بالقيمة e^{-4}).

10. باستخدام طريقة الرمي وطريقة الفروق المحدودة، أوجد حل مسألة القيم الحدية التالية:

$$\frac{d^2 y}{dx^2} + y = 0 \quad , \quad y(0) = 1 \quad , \quad y\left(\pi/2\right) = 1$$

أوجد الحل التحليلي وقارن(استعمل قيمة مناسبة لـ h).