7	بينما ($i \le n$ و $FLAG2 = 1$) نفذ الخطوة 8. (تجمع الأعداد الوسيطية).
8	إذا كان $\lvert x_i \rvert < y$ فضع $SUM = SUM + x_i^2$ $i = i + 1$ ما عدا ذلك ضع $FLAG2 = 0$ (نتج عدد كبير).
9	إذا كان $DONE = 0$ فإنه إذا كان $i > n$ فضع $NORM = (SUM)^{1/2}$ $DONE = 1$ ما عدا ذلك ضع $SUM = ((SUM)s)s$ (مقياس للأعداد الكبيرة). $FLAG3 = 1$
10	بينما ($i \le n$ و $FLAG3 = 1$) نفذ الخطوة 11.
11	ضع $SUM = SUM + (sx_i)^2$ (جمع الأعداد الكبيرة). $i = i + 1$
12	إذا كان $DONE = 0$ فإنه إذا كان $SUM^{1/2} < \lambda s$ فضع $NORM = (SUM)^{1/2}/s$ $DONE = 1$ ما عدا ذلك ضع $SUM = \lambda$ (المعيار كبير جدًا).
13	إذا كان $DONE = 1$ فالمخرجات ('Norm is', $NORM$) وما عدا ذلك المخرجات ('Norm ≥', $NORM$ حدث تخطٍ).
14	توقف.

لقد اختيرت العلاقات بين مؤشرات الآلة $emin, emax$ ،t, σ, λ ووسيطت الخوارزمية N, s, S, y, Y في [Brow, W, p. 471] على الصورة الآتية:

$N = 10^{e_N}$ بحيث $e_N = \lfloor (t-2)/2 \rfloor$، أكبر عدد صحيح أقل أو يساوي $(t-2)/2$.

$s = 10^{e_s}$ بحيث $e_s = \lfloor -(emax + e_N)/2 \rfloor$.

$S = 10^{e_S}$ بحيث $e_S = \lceil (1 - emin)/2 \rceil$، أصغر عدد صحيح أكبر أو يساوي $(1 - emin)/2$.

$y = 10^{e_y}$ بحيث $e_y = \lceil (emin + t - 2)/2 \rceil$.

$Y = 10^{e_Y}$ بحيث $e_Y = \lfloor (emax - e_N)/2 \rfloor$.

إن موثوقية هذه الخوارزمية قد زادت التعقيد كثيرًا إذا ما قُورنت بالخوارزمية التي بحثت في هذا الفصل. هناك أشكال كثيرة من البرمجيات العددية ذات الغرض العام متاحة تجاريًا وفي متناول الجمهور. إن معظم البرمجيات المبكرة كانت قد كتبت للحواسيب المركزية. وإليك مرجع جيد هو *Sources and Development of Mathematical Software* ومحرره Wayne Cowell [Co]. وفي الوقت الحاضر وحين أصبح الحاسوب ذو الشاشة قويًا بما يكفي. فقد أصبحت البرمجيات العددية متاحة للحواسيب الشخصية ومحطات العمل. وقد كتبت معظم هذه البرمجيات بلغة FORTRAN 77 على الرغم من كتابة بعضها بلغات Java ،FORTRAN 90 ،C++ ،C. إن عمليات ALGOL كانت قد قدمت لحسابات المصفوفات في عام 1971 [WR]. ثم طوّرت حقيبة برمجيات FORTRAN مبنية على عمليات ALGOL. وكان هذا التطوير موجّهًا إلى برمجيات EISPACK. لقد وُثّقت هذه البرمجيات عن طريق Springer – Verlag بوصفها جزءًا من مذكرات المحاضرات

لقد طُوّر الحاسوب الشخصي في أول الثمانينيات من القرن العشرين على يد ستيف وزنياك Steve Wozniak وستيف جوبس Steve Jobs مؤسسي حاسوب أبل Apple Compute.

وكان أول حاسوب محمول هو أوزبورن Osborne I الذي صنع عام 1981، مع أنه كان أكبر وأثقل كثيرا مما نتصوره الآن حاسوبا محمولا

إن نظام فورتران (FORmula TRANslator) كان لغة البرمجة العلمية الأصلية ذات الغرض العام. وما زالت قيد الاستخدام في الحالات التي تتطلب حسابات علمية متعمقة. وإن الطبعة الحالية المقننة لهذه اللغة هي FORTRAN

إن مشروع EISPACK هو أول حقيبة كبيرة للبرمجيات العددية التي أصبحت متاحة للاستخدام. وفتحت الطريق لبرمجيات أخرى تتبعها.

تأسست هندسة البرمجيات كنظام مخبري في سبعينيات وثمانينيات القرن العشرين حيث طور برنامج EISPACK في مختبرات آرغنون ولينباك فيما بعد. ومع بداية ثمانينيات القرن العشرين كانت مختبرات أرغنون ذات شهرة عالمية بكونها القائدة في هذا المجال ليس فقط في المجال الرمزي وإنما في مجال الحسابات العددية أيضا

في سلسلة علم الحاسوب Lecture Notes in Computer Science Series [Sm B] and [Gar]. تستخدم برمجيات FORTRAN لحساب القيم المميّزة eigenvalues والمتجهات المميّزة eigenvectors لعدد من أنواع المصفوفات المختلفة. إن EISPACK مصان من قبل نتلب netlib. ويمكن الدخول إليه عن طريق موقع نتلب http://www.netlib.org أما LINPACK فهو حقيبة برمجيات FORTRAN لتحليل نظم صيغ خطية وحلها وحل مسائل مربعات الصغرى الخطية. إن توثيق هذه البرمجية موجود في [DBMS]. وموضوع على موقع نتلب (netlib). وهناك مقدمة متدرجة خطوة خطوة لبرامج LINPACK و EISPACK و BLAS (Basic Linear Algebra Subprograms) مشروحة في [CV] ولقد أتيحت برمجية LAPACK أول مرة عام 1992. وهي من مكتبة برمجيات فورتران. وهي أقوى من LINPACK و EISPACK وذلك بتكامل هاتين المجموعتين من الخوارزميات في برمجية موحدة متجددة. وقد أعيد إنشاء البرمجيات للتوصل إلى معالجات المتّجهات على نحو كاف. وأداء أكثر كفاءة أو ذاكرة مشتركة. لقد طوّر LAPACK أفقيًا وبعمق في الطبعة 3.0 المتاحة JAVA ،C++ ،C ،FORTRAN 90 ،FORTRA. إن حقيبة BLAS ليست جزءًا من LAPACK. ولكن الشيفرة لـ BLAS موزعة مع LAPACK. وإن دليل LAPACK الطبعة الثالثة متاح من SIAM. The LAPACK User's Guide, 3rd ed.[AN] وهو متاح أيضا على الموقع

http://www.netlib.org / lapack /lug/lapack_lug.html

ويمكن الحصول على LAPACK أو برمجيات فردية من LAPACK عن طريق موقع نتلب netlib. وتوجد حقائب أخرى لحل أنواع محددة من المسائل موجودة للاستخدام العام ومتاحة على نتلب. ويمكن الاستزادة من المعلومات في المقالة "Software Distribution Using Netlib" للمؤلفين Dongarra و Roman و Wade [DRW]. إن هذه البرامج قادرة على فحص الشروط الخاصة جميعها التي يمكن أن تؤدي إلى الخطأ أو الفشل.

لقد أصبّحت IMSL أول مكتبة علمية للحواسيب العامة على نطاق واسع في عام 1970م. ومنذ ذلك الوقت أصبحت المكتبات متاحة للأنظمة الحاسوبية على مدى الحواسيب الشخصية.

سنناقش في آخر كل فصل بعض البرمجيات الملائمة للأغراض العامة. إن البرمجيات التجارية المتاحة تمثّل ما توصل إليه العلم في الطرائق العددية. وغالبًا ما تبنى محتوياتها على حقائب المستوى العمومي. ولكنها تحتوي على طرائق في مكتبات لأي نوع من المسائل تتكون من المكتبات STAT، MATH، و SFUN للرياضيات العددية والإحصاء والدّوال الخاصة على التوالي. تحتوي هذه المكتبات على أكثر من 900 برمجية كانت متاحة أصلا في FORTRAN 77 ومتاحة الآن في FORTRAN 90، C و Java. إن هذه البرمجيات تحل أكثر مسائل التحليل العددي شيوعًا. وإن المعلومات عن المكتبات متاحة على http://www.vni.com وهي متوفرة على نحو كاف وبتوثيق موسّع. ويوجد مثال برامجي لكل برنامج، بالإضافة إلى معلومات عن القاعدة المرجعية. ويحتوي IMSL على طرائق للأنظمة الخطية، تحليل نظام القيم، الاستيفاء الداخلي، التكامل والتفاضل، الصيغ التفاضلية، التمويلات، الصيغ غير الخطية، الأعظمية، وعمليات المصفوفة/المتجهات الرئيسة. وتحتوي المكتبة على برمجيات إحصائية واسعة أيضا. لقد وجدت مجموعة الخوارزميات العددية The Numerical Algorithms Group (NAG) في المملكة المتحدة منذ 1970. وتقدم NAG أكثر من 1000 برمجية في مكتبة FORTRAN 77 و400 برمجية في مكتبة C، وأكثر من 200 برمجية في مكتبة FORTRAN 90 ومكتبة للآلات المتوازية ومجموعات من محطات العمل أو الحواسيب الشخصية. إن مجموعة جزئية من مكتبة FORTRAN 77 (the NAG Foundation Library) متاحة للحواسيب الشخصية ومحطات العمل في حال كان فضاء العمل فيها محدودًا. يحتوي دليل استخدام NAG على تعليمات وأمثلة مع مثال

تأسست مجموعة الخوارزميات العددية (NAG) في المملكة المتحدة عام 1971م. وطوّرت أول مكتبة لبرمجة الرياضيات وهي تحوي الآن أكثر من 10 000 مستخدم. وتحوي أكثر من ألف دالّة في الرياضيات والإحصاء. تمتد من البرمجيات الإحصائية، والرمزية، المرئية والمحاكاة العددية إلى المجمّعات وأدوات التطوير التطبيقي

مخرجات لكل برمجية. إن [Ph] مرجع لقدمة مفيدة لبرمجيات NAG. تحتوي مكتبة NAG على برمجيات لتنفيذ معظم المهام الرئيسة في التحليل العددي بطريقة تشبه تلك التي في IMSL، وهي تحتوي على بعض البرمجيات الإحصائية أيضًا، ومجموعة من برمجيات الرسم. والمكتبة متاحة تجاريًا من مجموعة الخوارزميات العددية ذات الموقع على الشبكة http://www.nag.com لقد صُمّمت حقائب IMSL و NAG لعالم الرياضيات. والعالم أو المهندس الذي يرغب في استدعاء برمجيات Java ، C ، أو FORTRAN ذات النوعية العالية من داخل البرنامج. إن التوثيق المتاح مع الحقائب التجارية يشرح البرنامج المطلوب لاستخدام البرامج المكتبية. إن الحقائب الثلاث الآتية ذات بيئات فردية. وعندما تنشط فإن المستخدم يدخل أوامر تؤدي إلى حل مسألة ما عن طريق الحقيبة. وعلى كل حال فإن كل حقيبة تسمح بإنشاء برنامج ضمن لغة الأوامر فيها. إن MATLAB مختبر مصفوفي كان أصلًا برنامج FORTRAN نشره كليف مولر Cleve Moler [Mo] وقد بني معظم المختبر على برمجيات EISPACK و LINPACK مع أن دوال مثل الأنظمة غير الخطية، التكامل العددي، الشرائح التكعيبية، مطابقة المنحنيات، الأعظمية، الصيغ التفاضلية العادية، وأدوات الرسم قد ضمّنت فيه. إن هذا النظام القوي ذا الشمولية الذاتية مفيد خصوصًا لاستخدامه في مقرر الجبر الخطي التطبيقي. لقد أصبح ماتلاب MATLAB متاحًا منذ 1985، ويمكن الحصول على معلومات عن هذا النظام من شركة الأعمال الرياضية The MathWorks Inc، وعنوانها على الإنترنت هو http://www.mathworks.com والحقيبة الأخرى هي مابل Maple. وهي نظام حاسوبي جبري (CAS) طوّر في عام 1980م من قبل مجموعة الحساب الرمزي في جامعة واترلو Symbolic Computational Group at University of Waterloo إن تصميم نظام مابل Maple قد نُشرَ في بحث تشار، جيدس، جنتلمن، وجونت
B.W. Char, K.O. Geddes, W.M. Gentlemen, and G.H. gonnet [CGGG]

إن مابل Maple متاح منذ الثمانينيات 1980، وعنوان الحقيبة http://www.maplesoft.com ومابل Maple المكتوبة بلغة C قابلة لمعالجة المعلومات بطريقة رمزية. وإن هذه المعالجة الرمزية تسمح للمستخدم بالحصول على الأجوبة الدقيقة بدلًا من القيم العددية. وبإمكان مابل Maple إعطاء أجوبة دقيقة لمسائل رياضية، مثل التكاملات، الصيغ التفاضلية، والأنظمة الخطية. إنها تحوي إنشاءً برامجيًا. وتسمح بحفظ نص الأوامر في ملفات صحائف العمل التي يمكن إدخالها في مابل، ومن ثم تنفيذ الأوامر. لقد اختير مابل لاستخدامه في هذا الكتاب بسبب خصائص الحساب الرمزي، الحساب العددي وصحائف العمل. (تستخدم أوامر مابل وتكتب في متن هذا الكتاب). والحقيبة الثالثة هي Mathematical التي طوّرها ولفرام ريسيرج Wolfram Research عام 1985. ونشرت أول مرة عام 1988. إنها حقيبة قوية ومرغوبة من نوع CAS، وهي شائعة في مجالي التربية والأعمال. ويمكن الحصول على المعلومات حول هذه الحقيبة على العنوان http://www.wolfram.com. ويمكن الرجوع إلى كتب كودي ويت [Cody-Waite CW] وكوكلر [Kockler Ko] لمعلومات إضافية حول البرامج ومكتبات البرامج، وإلى بحث دوناكرا واكلر المنشور عام 1995 Dongarra - Walker ويمكن الرجوع إلى كتاب جايتيني-جاتلين وفرايز [CF] Chaitini - Chatelin and Frayse، وكذلك إلى بحث جولدبرغ [Goldberg Goldb] لمعلومات إضافية حول حسابات النقطة العائمة. إن كتب شندل [Schendell Sche]، فيليبس وفريمان [PF] Philips and Freeman، وجولب وأورتيقا [Golub Ortega GO] من الكتب التي تعرض تطبيق الطرائق العددية على الحواسيب المتوازية.

كُتبت ماتلاب MATLAB في الأصل لإتاحة الوصول إلى برمجية للمصفوفة في مشروعات لنباك وايزباك LINPACK و EISPACK

كُتب النموذج الأول في أواخر 1970 لاستخدامه في مقررات مبرهنة المصفوفات، الجبر الخطي والتحليل العددي. ويوجد في الوقت الحاضر أكثر من 500,000 مستخدم للماتلاب MATLAB في أكثر من 100 بلد

إن برمجيات NAG متوافقة مع مابل Maple ابتداءً من النموذج 7.0.

مع اختيارنا Maple نظامًا معياريًا لنا في CAS. فإن Mathematical المشهورة التي ظهرت في عام 1988 يمكن استخدامها لهذا الغرض.

الباب

2

حلول المعادلات بمتغير واحد

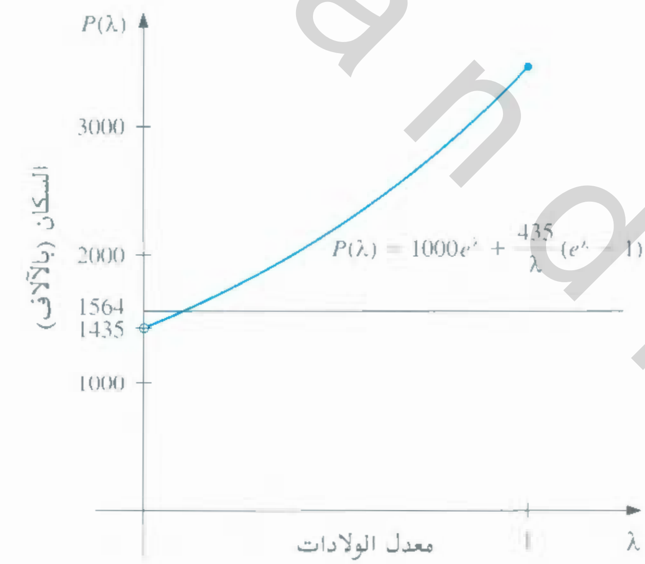
Solution of Equations in One Variable

مقدمة

يمكن نمذجة النمو السكاني عبر فترات زمنية قصيرة بافتراض أن النمو السكاني يكون متصلاً وبمعدل نمو يتناسب مع العدد الفعلي للسكان في الوقت المعين. فلو افترضنا أن $N(t)$ يساوي عدد السكان عند الزمن t. و λ يساوي معدل الولادات (ثابت) للسكان. فيمكن التعبير عن النمو السكاني بالصيغة التفاضلية

$$\frac{dN(t)}{dt} = \lambda N(t)$$

وحل هذه الصيغة هو $N(t) = N_0 e^{\lambda t}$. حيث تمثّل N_0 عدد السكان الابتدائي



وهذا النموذج الأسّي يكون مطابقا للواقع فقط عندما يكون المجتمع محصنا دون هجرة. وإذا ما سمحنا بمعدل هجرة v فإنّ الصيغة التفاضلية أعلاه تصبح

$$\frac{dN(t)}{dt} = \lambda N(t) + v$$

وحلها هو
$$N(t) = N_0 e^{\lambda t} + \frac{v}{\lambda}(e^{\lambda t} - 1)$$

ليكن مجتمع ما بعدد سكان ابتدائي قدره 1,000,000 نسمة، وأن 435,000 مهاجر انضموا إلى هذا المجتمع خلال السنة الأولى، ووُجد في نهاية السنة الأولى أن تعداد المجتمع هذا كان 1,564,000 نسمة. ولتحديد معدل ولادات هذا المجتمع، علينا تحديد قيمة λ من الصيغة
$$1{,}564{,}000 = 1{,}000{,}000 e^{\lambda} + \frac{435{,}000}{\lambda}(e^{\lambda} - 1)$$
إن الطرائق العددية التي نقدمها في هذا الباب تستخدم في تقريب الحلول لمثل هذه الصيغ، في الحالات التي يتعذر إيجاد حلول لها بالطرائق الجبرية. وسنتناول حل هذه الصيغة بعينها في التمرين (24) من الفصل (3.2).

1.2 طريقة التنصيف The Bisection Method

سنتناول في هذا الفصل واحدة من المسائل الرئيسة للتقريب العددي، وهي مسألة إيجاد جذور الصيغ. ويتضمن هذا الأسلوب إيجاد جذر أو حل لصيغة بالصيغة $f(x) = 0$، للدّالة f، ويسمّى جذر هذه الصيغة صفرًا للدّالة f أيضًا.

إن مسألة إيجاد تقريب لجذور الصيغة يعود إلى ما قبل العام 1700 قبل الميلاد. فقد كان من ضمن مجموعة Yale البابلية جدول بالرموز المسمارية يعود إلى تلك الحقبة، ويعطي عددًا ذا الأساس 60، ويعادل العدد 1,414222، كقيمة تقريبية للجذر $\sqrt{2}$، وهي تمهيدية تقترب دقتها إلى حد 10^{-5}. ويمكن إيجاد هذا التقريب بتطبيق أسلوب تناولناه في التمرين (19) من الفصل (2.2).

كما يشير اسمها "ثنائية التجزئة"، هــو التقسـيم (في هذه الحالـة هي الفـترة) لجزأين (في هذه الحالــة متكافئين) اثنين.

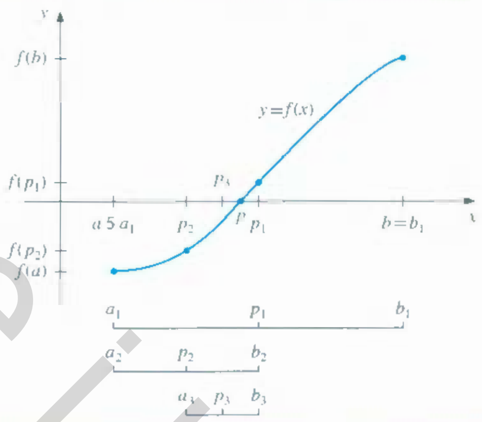
يعتمد الأسلوب الأول على مبرهنة القيمة الوسطية، ويسمّى "طريقة التنصيف". وهنا نفترض أن f تمثّل دالّة متصلة ومعرّفة على الفترة $[a, b]$ مع كون كل من الدّوال $f(a)$ و $f(b)$ ذات إشارة مختلفة. ووفقًا لمبرهنة القيمة الوسطية، فمن المؤكد وجود العدد p في الفترة (a, b) بحيث يكون $f(p) = 0$. وعلى الرغم من أن العملية قائمة على وجود عدد من الجذور يزيد عن واحد في الفترة (a, b) فإنّنا – ولغرض التبسيط – نفترض وحدانية الجذر. وتستدعي الطريقة تكرار تنصيف الفترات الجزئية للفترة $[a, b]$، وتحديد النصف الذي يحتوي p في كل خطوة.

في علم الحاسوب، فإن عملية تقسـيم مجموعة باستمرار إلى نصفين للبحث عن حل للمســألة – كما تفعل طريقة التنصيف – تُعرف بأنها عملية البحث الثنائي binary.

ليكن $a_1 = a$ و $b_1 = b$ في البداية، وليمثل p_1 منتصف الفترة $[a, b]$. أي أن
$$p_1 = a_1 + \frac{b_1 - a_1}{2} = \frac{a_1 + b_1}{2}$$
إذا كانت $f(p_1) = 0$ فإنّ $p = p_1$، وبهذا نكون قد انتهينا من الحل. أما إذا كانت $f(p_1) \neq 0$ فإنّ إشارة $f(p_1)$ هي إما إشارة $f(a_1)$ أو إشارة $f(b_1)$. فإذا كانت $f(p_1)$ و $f(a_1)$ لهما الإشارة نفسها، تكون $p \in (p_1, b_1)$ ونعطي القيم $a_2 = p_1$ و $b_2 = b_1$. أمّا إذا كانت $f(p_1)$ و $f(a_1)$ لهما إشارتان مختلفتان، فعندها تكون $p \in (a_1, p_1)$، ونعطي القيم $a_2 = a_1$ و $b_2 = p_1$. وعندئذ

نعيد تكرار الخطوة على الفترة $[a_2,b_2]$. وبهذا نحصل على الطريقة الموضحة في الخوارزمية (1.2). (انظر شكل 1.2).

شكل 1.2



ALGORITHM

الخوارزمية 1.2

التنصيف Bisection

لإيجاد حل لـ $f(x) = 0$ للدالّة f المتصلة على الفترة $[a, b]$ مع اختلاف إشارة $f(a)$ و $f(b)$.

المدخلات: نقاط النهاية a, b الحدود المسموح بها TOL. أكبر عدد لمرات التكرار N_0.

المخرجات: حل تقريبي p أو عبارة "فشل".

الخطوة	المضمون
1	ضع $i = 1$، $FA = f(a)$.
2	ما دام $i \le N_0$ فنفّذ الخطوات 3 – 6 الآتية.
3	ضع $p = a + (b - a)/2$ (احسب P_i) $FP = f(p)$.
4	إذا حصل $FP = 0$، أو $(b - a)/2 < TOL$. المُخرج (p) (استكملت العملية بنجاح). توقف.
5	ضع $i = i + 1$.
6	إذا حصل $FA \cdot FP > 0$ فضع $a = p$ (واحسب a_i, b_i). $FA = FP$ وخلاف ذلك ضع $b = p$.
7	المُخرجات (فشلت الطريقة بعد N_0 من التكرارات و N_0، $N_0 =$ ˙) توقف.

■

يمكن تطبيق عمليات إيقاف أخرى في الخطوة (4) من الخوارزمية (1.2) أو في أي أسلوب

تكرار ضمن هذا الباب. ويمكننا على سبيل المثال اختيار حد السماح $\varepsilon > 0$ وتوليد $p_1, \ldots, p_N$ لتحقّق أحد الشروط الآتية:

$$|p_N - p_{N-1}| < \varepsilon \tag{1.2}$$

$$\frac{|p_N - p_{N-1}|}{|p_N|} < \varepsilon, \quad p_N \neq 0 \tag{2.2}$$

أو

$$|f(p_N)| < \varepsilon \tag{3.2}$$

وتجدر الإشارة هنا إلى بعض الصعوبات التي يمكن أن تظهر عند تطبيق أي من قواعد الإيقاف هذه. فعلى سبيل المثال يمكن أن تؤول نهاية المتتالية $\{p_n\}_{n=0}^{\infty}$ مع خاصية الفروقات $p_n - p_{n-1}$ إلى الصفر. حيث ليس للمتتالية نفسها نهاية محدودة. (انظر تمرين 17). ومن المحتمل أن تقترب $f(p_n)$ من الصفر أيضا. حيث تختلف p_n منطقيًا عن p. (انظر تمرين 16) تعدّ المتباينة (2.2) أفضل قاعدة إيقاف، لكونها تقترب من اختبار الخطأ النسبي دون أي معلومات إضافية حول f أو p.

وعند استخدام الحاسوب لتوليد التقريبات، فمن الأفضل تحديد الحد الأعلى لعدد مرات التكرار. وهذا من شأنه استبعاد إمكانية إدخال عدد لانهائي من التدويرات. وهي حالة يمكن حدوثها عندما لا يكون للمتتالية نهاية محدودة (عند ترميز البرنامج ترميزا خطأً أيضًا). وقد تضمنت الخطوة (2) من الخوارزمية (1.2) ذلك. حيث حدّدنا N_0 مع إيقاف العملية عندما يكون $i > N_0$.

لاحظ أنه عند البدء بخوارزمية التنصيف. يتعين إيجاد الفترة $[a, b]$ مع $f(a) \cdot f(b) < 0$. ومن المعلوم في كل خطوة أن مدى الفترة المتضمن صفرا لـ f ينقص بمقدار 2، لذا يفضل اختيار الفترة الابتدائية $[a, b]$ أصغر ما يمكن. وعلى سبيل المثال إذا كانت $f(x) = 2x^3 - x^2 + x - 1$ يكون لدينا كل من

$$f(-4) \cdot f(4) < 0 \quad \text{و} \quad f(0) \cdot f(1) < 0$$

وبناءً عليه فإنّ خوارزمية التنصيف يمكن استخدامها لأي من الفترتين $[-4, 4]$ أو $[0, 1]$ وعلى أي حال فإنّ استخدام الفترة $[0, 1]$ بدلًا من $[-4, 4]$ سيقلّص عدد مرات التكرار المطلوبة إلى 3 مرات، للوصول إلى الدقة المحددة. ويوضح المثال الآتي خوارزمية التنصيف. وتتوقف عملية التكرار في هذا المثال عندما يكون الخطأ النسبي أقل من 0.0001، أي عندما نكون

$$\frac{|p - p_n|}{|p|} < 10^{-4}$$

مثال 1 الصيغة $f(x) = x^3 + 4x^2 - 10 = 0$ لها جذر ضمن $[1, 2]$، لأن $f(1) = -5$ و $f(2) = 14$ وتعطي خوارزمية التنصيف القيم في جدول (1.2). ونلاحظ بعد 13 تكرارًا أن $p_{13} = 1.365112305$ يقرب الجذر p بخطأ مقداره

$$|p - p_{13}| < |b_{14} - a_{14}| = |1.365234375 - 1.365112305| = 0.000122070$$

ولكون $|a_{14}| < |p|$ فإن $\dfrac{|p - p_{13}|}{|p|} < \dfrac{|b_{14} - a_{14}|}{|a_{14}|} \leq 9.0 \times 10^{-5}$

وبناءً عليه فإنّ التقريب إلى أربع منازل عشرية على الأقل صحيح. والقيمة الصحيحة لـ p لتسع منازل هي 1.365230013. لاحظ أن p_9 تكون أقرب إلى p من التقريب الأخير p_{13}.

جدول 1.2

n	a_n	b_n	p_n	$f(p_n)$
1	1.0	2.0	1.5	2.375
2	1.0	1.5	1.25	−1.79687
3	1.25	1.5	1.375	0.16211
4	1.25	1.375	1.3125	−0.84839
5	1.3125	1.375	1.34375	−0.35098
6	1.34375	1.375	1.359375	−0.09641
7	1.359375	1.375	1.3671875	0.03236
8	1.359375	1.3671875	1.36328125	−0.03215
9	1.36328125	1.3671875	1.365234375	0.000072
10	1.36328125	1.365234375	1.364257813	−0.01605
11	1.364257813	1.365234375	1.364746094	−0.00799
12	1.364746094	1.365234375	1.364990235	−0.00396
13	1.364990235	1.365234375	1.365112305	−0.00194

قد نستغرب حدوث ذلك؛ لأن $|f(p_9)| < |f(p_{13})|$، ولكن ليس بالإمكان التأكد منه إلا عند معرفة الجواب الصحيح. ■

وعلى الرغم من وضوح مفهوم طريقة التنصيف، إلا أنها تتضمن عيوبًا ملموسة؛ فهي بطيئة في تقاربها (أي أن N قد تكون كبيرة قليلًا قبل الحصول على قيمة صغيرة ملموسة لـ $|p - p_N|$)، بالإضافة إلى إمكانية تجاهل وسيط جيد للمقاربة دون الوقوف عنده. وعلى الرغم من ذلك، فإنّ الطريقة هذه تتسم بالجودة؛ كونها تتقارب إلى حل محدود دائمًا، ولهذا السبب فهي تستخدم بداية في طرائق أخرى أكثر جدوى غالبًا، وسنتناولها في آخر هذا الباب.

مبرهنة 1.2 ليكن $f \in C[a, b]$ و $f(a) \cdot f(b) < 0$، عندئذ تولد طريقة التنصيف المتتالية $\{p_n\}_{n=1}^{\infty}$ التي تقترب من الجذر p للدالّة f حيث

$$|p_n - p| \le \frac{b-a}{2^n} \quad \text{عندما} \quad n \ge 1$$ ■

البرهان لكل $n \ge 1$ يكون لدينا

$$b_n - a_n = \frac{1}{2^{n-1}}(b-a) \quad \text{و} \quad p \in (a_n, b_n)$$

وبما أن $p_n = \frac{1}{2}(a_n + b_n)$ لكل $n \ge 1$ فإنّ ذلك يؤدي إلى

$$|p_n - p| \le \frac{1}{2}(b_n - a_n) = \frac{b-a}{2^n}$$ ■ ■ ■

ولأن

$$|p_n - p| \le (b-a)\frac{1}{2^n}$$

فإنّ المتتالية $\{p_n\}_{n=1}^{\infty}$ تتقارب نحو p مع معدل تقارب $O(1/2^n)$. أي أن

$$p_n = p + O\left(\frac{1}{2^n}\right)$$

ومن الضروري إدراك أن المبرهنة (1.2) تعطي حدًّا فقط لخطأ التقريب، وربما يكون هذا الحد ليس