Some Perturbation Problems 02-08-17

N. T. Gladd

Initialization: Be sure the files *NTGStylesheet2.nb* and *NTGUtilityFunctions.m* is are in the same directory as that from which this notebook was loaded. Then execute the cell immediately below by mousing left on the cell bar to the right of that cell and then typing “shift” + “enter”. Respond “Yes” in response to the query to evaluate initialization cells.

In[88]:=
```
SetDirectory[NotebookDirectory[]];
(* set directory where source files are located *)
SetOptions[EvaluationNotebook[], (* load the StyleSheet *)
 StyleDefinitions → Get["NTGStylesheet2.nb"]];
Get["NTGUtilityFunctions.m"]; (* Load utilities package *)
```

Purpose

Perturbation methods are broadly useful but often require long repetitive calculations that can be facilitated by the use of Mathematica. Moreover, having immediate access to visualization and numerical methods provides guidance when formulating analytical calculations as well as convenience when checking the accuracy and range of validity of results.

In this notebook, I work through some representative perturbation problems related to algebraic equations in detail. Such problems provide a foundation for the broader application of perturbation methods to differential equations, integrals, and more.

The book *Perturbation Methods*, E. J. Hinch, provides a quite readable introduction to perturbation methods — a topic for which there is a vast literature. Most of the problems in this notebook are drawn from Hinch. Other favorite books are *Introduction to Asymptotics and Special Functions*, F. W. J. Olver, *Advanced Mathematical Methods for Scientists and Engineers*, C. M. Bender and S. A Orszag, *Introduction to the Foundations of Applied Mathematics*, Mark H. Holmes, and the books by A. H. Nayfeh. On a personal note, I attended and much enjoyed lectures given by Frank Olver, when I was a research faculty member at the University of Maryland in the late 1970s.

The problems treated below are well described in the sources from which they are drawn. My intent is not to repeat the existing exposition but to illustrate how the use of Mathematica can facilitate the solution process.

The sections below treat

- 1 Iteration and expansion methods for a regular perturbation problem
- 2 A singular perturbation problem
- 3 A problem involving expansion in fractional powers of ϵ
- 4 Another method for determining an appropriate expansion
- 5 A problem for which iteration is preferred

1 Iteration and expansion methods for a regular perturbation problem

In Section 1.1 Hinch considers the model equation

```
In[5]:=  w1[1] = x^2 + ϵ x - 1 == 0
Out[5]=  -1 + x^2 + x ϵ == 0
```

with $0 < \epsilon << 1$. If ϵ multiplied the term with the highest degree (x^2) this would be a singular perturbation problem. Otherwise it is a regular perturbation problem.

Of course, this model equation is immediately solvable

```
In[6]:=  w1[2] = Solve[w1[1], x]
```

$$\text{Out[6]= } \left\{\left\{x \to \frac{1}{2}\left(-\epsilon - \sqrt{4+\epsilon^2}\right)\right\}, \left\{x \to \frac{1}{2}\left(-\epsilon + \sqrt{4+\epsilon^2}\right)\right\}\right\}$$

1a Iteration

Hinch suggests that the following rearrangement is suitable for application of the iteration method

```
In[10]:= w1a[1] = Solve[w1[1] /. x^2 → x2, x2][[1, 1]] /. x2 → x^2 // RE
Out[10]= x^2 == 1 - x ϵ
```

The lowest order solution assumes $\epsilon = 0$.

```
In[11]:= w1a[2] = Solve[w1a[1] /. ϵ → 0, x]
Out[11]= {{x → -1}, {x → 1}}
```

I follow Hinch and consider the development of the iteration solution of the positive branch ($x \to 1$ as $\epsilon \to 0$)

```
In[14]:= w1a[3] = (√#) & /@ w1a[1] // PowerExpand
```

Out[14]= $x == \sqrt{1 - x \epsilon}$

The iteration scheme is

```
In[15]:= w1a[4] = (w1a[3][[1]] /. x → x_{n+1}) == (w1a[3][[2]] /. x → x_n)
```

Out[15]= $x_{1+n} == \sqrt{1 - \epsilon\, x_n}$

and the starting point is

```
In[16]:= w1a[5] = x_0 == 1
```

Out[16]= $x_0 == 1$

An analogous scheme could be established for the negative branch.

A recursion that implements this iteration scheme is easy to establish.

```
In[20]:= Clear[X];
X[0] = 1;
X[n_] := X[n] = √(1 - ϵ X[n - 1])
```

Some comments on this implementation are

1) I used X instead of x so that the earlier formulae would not be “contaminated” by establishing an explicit numerical value for x within the notebook environment.

2) I used X[n_] instead of $X_{n\_}$ because, as one Mathematica guru on StackExchange puts it, “Don’t use subscripts in calculations unless you are an expert.” The reason for this admonition is that subscripted variables in Mathematica are not interpreted as “Symbols” as intuition would dictate. You can get into trouble by not knowing exactly how Subscripts are implemented. Nonetheless, my background in theoretical physics often compels me to use subscript notation.

3) The somewhat mysterious line `X[n_] := X[n] = √(1 - ϵ X[n - 1])` implements “memoization,” a useful technique by which the values of previous calculations are remembered rather than recalculated (significant time savings).

It is not necessary to loop through the various values of n. Asking for a particular value, say X[6] triggers the recursive valuation of all earlier terms.

```
In[23]:= X[6]
```

Out[23]= $\sqrt{\left(1 - \epsilon \sqrt{\left(1 - \epsilon \sqrt{\left(1 - \epsilon \sqrt{\left(1 - \epsilon \sqrt{1 - \sqrt{1 - \epsilon}\ \epsilon}\right)}\right)}\right)}\right)}$

So, if I ask for the values associated with X

```
In[24]:= ?? X
```

```
Global`X
```

$X[0] = 1$

$X[1] = \sqrt{1-\epsilon}$

$X[2] = \sqrt{1-\sqrt{1-\epsilon}\ \epsilon}$

$X[3] = \sqrt{1-\epsilon\sqrt{1-\sqrt{1-\epsilon}\ \epsilon}}$

$X[4] = \sqrt{1-\epsilon\sqrt{1-\epsilon\sqrt{1-\sqrt{1-\epsilon}\ \epsilon}}}$

$X[5] = \sqrt{\left(1-\epsilon\sqrt{1-\epsilon\sqrt{1-\epsilon\sqrt{1-\sqrt{1-\epsilon}\ \epsilon}}}\right)}$

$X[6] = \sqrt{\left(1-\epsilon\sqrt{\left(1-\epsilon\sqrt{\left(1-\epsilon\sqrt{\left(1-\epsilon\sqrt{1-\sqrt{1-\epsilon}\ \epsilon}\right)}\right)}\right)}\right)}$

$X[n\_] := X[n] = \sqrt{1-\epsilon\, X[n-1]}$

For future use I generate a list of these results

In[25]:=
```
w1a[6] = Table[{n, X[n]}, {n, 0, 6}]
```

Out[25]=

$$\left\{\{0, 1\}, \left\{1, \sqrt{1-\epsilon}\right\}, \left\{2, \sqrt{1-\sqrt{1-\epsilon}\ \epsilon}\right\}, \left\{3, \sqrt{1-\epsilon\sqrt{1-\sqrt{1-\epsilon}\ \epsilon}}\right\},\right.$$
$$\left\{4, \sqrt{\left(1-\epsilon\sqrt{1-\epsilon\sqrt{1-\sqrt{1-\epsilon}\ \epsilon}}\right)}\right\}, \left\{5, \sqrt{\left(1-\epsilon\sqrt{\left(1-\epsilon\sqrt{\left(1-\epsilon\sqrt{1-\sqrt{1-\epsilon}\ \epsilon}\right)}\right)}\right)}\right\},$$
$$\left.\left\{6, \sqrt{\left(1-\epsilon\sqrt{\left(1-\epsilon\sqrt{\left(1-\epsilon\sqrt{\left(1-\epsilon\sqrt{1-\sqrt{1-\epsilon}\ \epsilon}\right)}\right)}\right)}\right)}\right\}\right\}$$

and pretty-print them with

In[26]:=
```
LGrid[Join[{{"n", "Xn"}}, w1a[6]], "Iteration Method"]
```

Out[26]=

Iteration Method

n	X_n
0	1
1	$\sqrt{1-\epsilon}$
2	$\sqrt{1-\sqrt{1-\epsilon}\,\epsilon}$
3	$\sqrt{1-\epsilon\sqrt{1-\sqrt{1-\epsilon}\,\epsilon}}$
4	$\sqrt{1-\epsilon\sqrt{1-\epsilon\sqrt{1-\sqrt{1-\epsilon}\,\epsilon}}}$
5	$\sqrt{\left(1-\epsilon\sqrt{\left(1-\epsilon\sqrt{\left(1-\epsilon\sqrt{1-\sqrt{1-\epsilon}\,\epsilon}\right)}\right)}\right)}$
6	$\sqrt{\left(1-\epsilon\sqrt{\left(1-\epsilon\sqrt{\left(1-\epsilon\sqrt{\left(1-\epsilon\sqrt{1-\sqrt{1-\epsilon}\,\epsilon}\right)}\right)}\right)}\right)}$

The repeated radicals are not particularly informative so I generate power series representations

In[27]:=
```
w1a[7] = ({#[[1]], Series[#[[2]], {ϵ, 0, 5}]}) & /@ w1a[6]
```

Out[27]=

$$\left\{\{0,1\},\left\{1,1-\frac{\epsilon}{2}-\frac{\epsilon^2}{8}-\frac{\epsilon^3}{16}-\frac{5\epsilon^4}{128}-\frac{7\epsilon^5}{256}+O[\epsilon]^6\right\},\left\{2,1-\frac{\epsilon}{2}+\frac{\epsilon^2}{8}+\frac{\epsilon^3}{8}+\frac{11\epsilon^4}{128}+\frac{3\epsilon^5}{64}+O[\epsilon]^6\right\},\right.$$
$$\left\{3,1-\frac{\epsilon}{2}+\frac{\epsilon^2}{8}-\frac{9\epsilon^4}{128}-\frac{5\epsilon^5}{64}+O[\epsilon]^6\right\},\left\{4,1-\frac{\epsilon}{2}+\frac{\epsilon^2}{8}-\frac{\epsilon^4}{128}+\frac{\epsilon^5}{32}+O[\epsilon]^6\right\},$$
$$\left.\left\{5,1-\frac{\epsilon}{2}+\frac{\epsilon^2}{8}-\frac{\epsilon^4}{128}+O[\epsilon]^6\right\},\left\{6,1-\frac{\epsilon}{2}+\frac{\epsilon^2}{8}-\frac{\epsilon^4}{128}+O[\epsilon]^6\right\}\right\}$$

In[28]:=
```
LGrid[Join[{{"n", "Xn"}}, w1a[7]], "Iterative Expansion"]
```

Out[28]=

Iterative Expansion

n	X_n
0	1
1	$1-\frac{\epsilon}{2}-\frac{\epsilon^2}{8}-\frac{\epsilon^3}{16}-\frac{5\epsilon^4}{128}-\frac{7\epsilon^5}{256}+O[\epsilon]^6$
2	$1-\frac{\epsilon}{2}+\frac{\epsilon^2}{8}+\frac{\epsilon^3}{8}+\frac{11\epsilon^4}{128}+\frac{3\epsilon^5}{64}+O[\epsilon]^6$
3	$1-\frac{\epsilon}{2}+\frac{\epsilon^2}{8}-\frac{9\epsilon^4}{128}-\frac{5\epsilon^5}{64}+O[\epsilon]^6$
4	$1-\frac{\epsilon}{2}+\frac{\epsilon^2}{8}-\frac{\epsilon^4}{128}+\frac{\epsilon^5}{32}+O[\epsilon]^6$
5	$1-\frac{\epsilon}{2}+\frac{\epsilon^2}{8}-\frac{\epsilon^4}{128}+O[\epsilon]^6$
6	$1-\frac{\epsilon}{2}+\frac{\epsilon^2}{8}-\frac{\epsilon^4}{128}+O[\epsilon]^6$

This series can be compared against a power series expansion of the exact solution. For the positive branch that is

In[29]:=
```
w1[2]⟦2, 1⟧
```
Out[29]=
$$x \to \frac{1}{2}\left(-\epsilon + \sqrt{4+\epsilon^2}\right)$$

In[30]:=
```
w1a[8] = x → Normal@Series[w1[2]⟦2, 1, 2⟧, {ϵ, 0, 5}]
```
Out[30]=
$$x \to 1 - \frac{\epsilon}{2} + \frac{\epsilon^2}{8} - \frac{\epsilon^4}{128}$$

which agrees with the iteration result at order 6.

In[31]:=
```
w1a[8]⟦2⟧ - w1a[7]⟦-1, 2⟧
```
Out[31]=
$$O[\epsilon]^6$$

I have focused on the calculation. See Hinch for discussion of merits and demerits of the Iteration method.

1b Expansion method

Hinch develops an approximation for the positive branch by assuming the solution is well represented by the expansion

In[32]:=
```
expansionRule = x → Sum[ϵ^i x_i, {i, 0, 7}]
```
Out[32]=
$$x \to x_0 + \epsilon x_1 + \epsilon^2 x_2 + \epsilon^3 x_3 + \epsilon^4 x_4 + \epsilon^5 x_5 + \epsilon^6 x_6 + \epsilon^7 x_7$$

In[33]:=
```
w1b[1] = w1[1] /. expansionRule
```
Out[33]=
$$-1 + \epsilon\left(x_0 + \epsilon x_1 + \epsilon^2 x_2 + \epsilon^3 x_3 + \epsilon^4 x_4 + \epsilon^5 x_5 + \epsilon^6 x_6 + \epsilon^7 x_7\right) + \left(x_0 + \epsilon x_1 + \epsilon^2 x_2 + \epsilon^3 x_3 + \epsilon^4 x_4 + \epsilon^5 x_5 + \epsilon^6 x_6 + \epsilon^7 x_7\right)^2 == 0$$

Collect terms at various orders of ϵ

In[34]:=
```
w1b[2] = CoefficientList[ExpandAll[w1b[1]⟦1⟧], ϵ]
```
Out[34]=
$$\{-1 + x_0^2,\ x_0 + 2 x_0 x_1,\ x_1 + x_1^2 + 2 x_0 x_2,\ x_2 + 2 x_1 x_2 + 2 x_0 x_3,$$
$$x_2^2 + x_3 + 2 x_1 x_3 + 2 x_0 x_4,\ 2 x_2 x_3 + x_4 + 2 x_1 x_4 + 2 x_0 x_5,\ x_3^2 + 2 x_2 x_4 + x_5 + 2 x_1 x_5 + 2 x_0 x_6,$$
$$2 x_3 x_4 + 2 x_2 x_5 + x_6 + 2 x_1 x_6 + 2 x_0 x_7,\ x_4^2 + 2 x_3 x_5 + 2 x_2 x_6 + x_7 + 2 x_1 x_7,$$
$$2 x_4 x_5 + 2 x_3 x_6 + 2 x_2 x_7,\ x_5^2 + 2 x_4 x_6 + 2 x_3 x_7,\ 2 x_5 x_6 + 2 x_4 x_7,\ x_6^2 + 2 x_5 x_7,\ 2 x_6 x_7,\ x_7^2\}$$

This leads to a sequence of equations

In[35]:=
```
w1b[3] = (# == 0) & /@ w1b[2];
w1b[3] // ColumnForm
```

Out[36]=

$$-1 + x_0^2 == 0$$
$$x_0 + 2\,x_0\,x_1 == 0$$
$$x_1 + x_1^2 + 2\,x_0\,x_2 == 0$$
$$x_2 + 2\,x_1\,x_2 + 2\,x_0\,x_3 == 0$$
$$x_2^2 + x_3 + 2\,x_1\,x_3 + 2\,x_0\,x_4 == 0$$
$$2\,x_2\,x_3 + x_4 + 2\,x_1\,x_4 + 2\,x_0\,x_5 == 0$$
$$x_3^2 + 2\,x_2\,x_4 + x_5 + 2\,x_1\,x_5 + 2\,x_0\,x_6 == 0$$
$$2\,x_3\,x_4 + 2\,x_2\,x_5 + x_6 + 2\,x_1\,x_6 + 2\,x_0\,x_7 == 0$$
$$x_4^2 + 2\,x_3\,x_5 + 2\,x_2\,x_6 + x_7 + 2\,x_1\,x_7 == 0$$
$$2\,x_4\,x_5 + 2\,x_3\,x_6 + 2\,x_2\,x_7 == 0$$
$$x_5^2 + 2\,x_4\,x_6 + 2\,x_3\,x_7 == 0$$
$$2\,x_5\,x_6 + 2\,x_4\,x_7 == 0$$
$$x_6^2 + 2\,x_5\,x_7 == 0$$
$$2\,x_6\,x_7 == 0$$
$$x_7^2 == 0$$

For n ≥ 1, the equations are linear and easily solved

In[37]:=
```
w1b[4] = Table[Solve[w1b[3][[i]], x_{i-1}][[1, 1]] , {i, 2, 8}]
```

Out[37]=

$$\left\{x_1 \to -\frac{1}{2},\ x_2 \to \frac{-x_1 - x_1^2}{2\,x_0},\ x_3 \to \frac{-x_2 - 2\,x_1\,x_2}{2\,x_0},\ x_4 \to \frac{-x_2^2 - x_3 - 2\,x_1\,x_3}{2\,x_0},\ x_5 \to \frac{-2\,x_2\,x_3 - x_4 - 2\,x_1\,x_4}{2\,x_0},\right.$$
$$\left. x_6 \to \frac{-x_3^2 - 2\,x_2\,x_4 - x_5 - 2\,x_1\,x_5}{2\,x_0},\ x_7 \to \frac{1}{2\,x_0}\left(-2\,x_3\,x_4 - 2\,x_2\,x_5 - x_6 - 2\,x_1\,x_6\right)\right\}$$

For the positive branch $x_0 = 1$

In[38]:=
```
w1b[5] = Join[{x_0 → 1}, w1b[4]]
```

Out[38]=

$$\left\{x_0 \to 1,\ x_1 \to -\frac{1}{2},\ x_2 \to \frac{-x_1 - x_1^2}{2\,x_0},\ x_3 \to \frac{-x_2 - 2\,x_1\,x_2}{2\,x_0},\right.$$
$$x_4 \to \frac{-x_2^2 - x_3 - 2\,x_1\,x_3}{2\,x_0},\ x_5 \to \frac{-2\,x_2\,x_3 - x_4 - 2\,x_1\,x_4}{2\,x_0},$$
$$\left. x_6 \to \frac{-x_3^2 - 2\,x_2\,x_4 - x_5 - 2\,x_1\,x_5}{2\,x_0},\ x_7 \to \frac{1}{2\,x_0}\left(-2\,x_3\,x_4 - 2\,x_2\,x_5 - x_6 - 2\,x_1\,x_6\right)\right\}$$

and

In[39]:=
```
w1b[6] = Join[w1b[5][[1 ;; 2]], Table[w1b[5][[i]] //. w1b[5][[1 ;; i - 1]], {i, 3, 8}] ]
```

Out[39]=

$$\left\{x_0 \to 1,\ x_1 \to -\frac{1}{2},\ x_2 \to \frac{1}{8},\ x_3 \to 0,\ x_4 \to -\frac{1}{128},\ x_5 \to 0,\ x_6 \to \frac{1}{1024},\ x_7 \to 0\right\}$$

Mathematica note: ReplaceRepeated //. is required instead of Replace /.

The expansion method solution is

```
w1b[6] = expansionRule /. w1b[6]
```

$$x \to 1 - \frac{\epsilon}{2} + \frac{\epsilon^2}{8} - \frac{\epsilon^4}{128} + \frac{\epsilon^6}{1024}$$

which agrees with the results from a power series expansion of the exact solution

2 A singular perturbation problem

Hinch 1.2 treats a singular perturbation problem

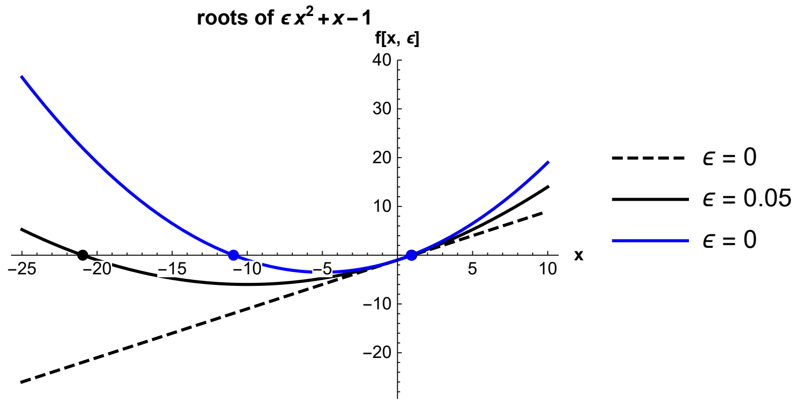
In[40]:=
```
w2[1] = ϵ x^2 + x - 1 == 0
```

Out[40]=
$$-1 + x + x^2 \epsilon == 0$$

Complications arise because ϵ multiplies the term with highest degree

A quick visualization indicates two roots that become widely separated as ϵ increases. For $\epsilon = 0$ the equation is linear.

In[41]:=
```
Module[{f, roots, points, inset, lab, legends, ϵVals = {0.05, 0.1}},
 f[x_, ϵ_] := ϵ x^2 + x - 1;
 roots = NSolve[f[x, #], x] & /@ ϵVals;
 points = Point[{x, 0}] /. roots;
 lab = Stl@StringForm["roots of ``", TraditionalForm[f[x, ϵ]]];
 legends = {StringForm["ϵ = 0"],
   StringForm["ϵ = ``", ϵVals [[1]] ], StringForm["ϵ = 0", ϵVals [[2]] ]};
 Plot[{f[x, 0], f[x, ϵVals [[1]] ], f[x, ϵVals [[2]]]}, {x, -25, 10},
  PlotStyle → {Directive[Black, Dashed], Black, Blue},
  AxesLabel → {Stl["x"], Stl["f[x, ϵ]"]},
  PlotLegends → legends, ImageSize → 300, PlotLabel → lab,
  Epilog → {PointSize[0.02], {Black, points [[1]]}, {Blue, points [[2]]}} ]]
```

Out[41]=


Solving the quadratic, the exact solutions are

In[42]:=
```
w2[2] = Solve[w2[1], x]
```

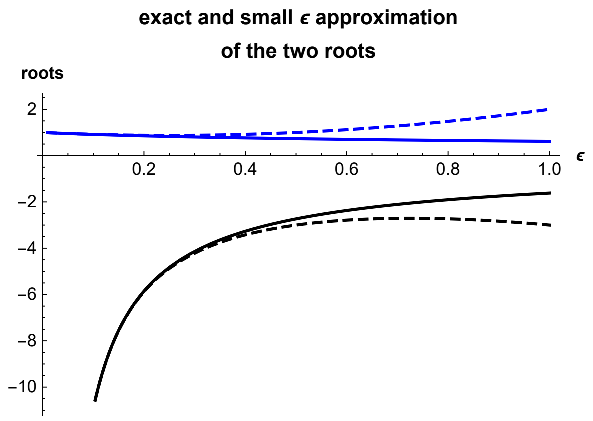
Out[42]=
$$\left\{\left\{x\to\frac{-1-\sqrt{1+4\,\epsilon}}{2\,\epsilon}\right\},\left\{x\to\frac{-1+\sqrt{1+4\,\epsilon}}{2\,\epsilon}\right\}\right\}$$

while the solutions for ϵ << 1 are

In[43]:=
```
w2[3] = (#[[1, 1]] → Normal@Series[#[[1, 2]], {ϵ, 0, 7}]) & /@ w2[2]
```

Out[43]=
$$\left\{x\to -1-\frac{1}{\epsilon}+\epsilon-2\,\epsilon^2+5\,\epsilon^3-14\,\epsilon^4+42\,\epsilon^5-132\,\epsilon^6+429\,\epsilon^7,\right.$$
$$\left. x\to 1-\epsilon+2\,\epsilon^2-5\,\epsilon^3+14\,\epsilon^4-42\,\epsilon^5+132\,\epsilon^6-429\,\epsilon^7\right\}$$

In[44]:=
```
Module[{},
 Plot[{(-1 - Sqrt[1 + 4 ϵ])/(2 ϵ), -1 - 1/ϵ + ϵ - 2 ϵ^2, (-1 + Sqrt[1 + 4 ϵ])/(2 ϵ), 1 - ϵ + 2 ϵ^2}, {ϵ, 0.01, 1},
  PlotStyle → {Black, Directive[Black, Dashed], Blue, Directive[Blue, Dashed]},
  AxesLabel → {Stl["ϵ"], Stl["roots"]}, ImageSize → 300,
  PlotLabel → Stl["exact and small ϵ approximation\nof the two roots"]]]
```

Out[44]=


I apply the expansion method for the root at large negative x.

In this case, the <u>assumed</u> form of the expansion is

In[45]:=
```
expansionRule = x → Sum[ϵ^i x_i, {i, -1, 7}]
```

Out[45]=
$$x\to\frac{x_{-1}}{\epsilon}+x_0+\epsilon\,x_1+\epsilon^2\,x_2+\epsilon^3\,x_3+\epsilon^4\,x_4+\epsilon^5\,x_5+\epsilon^6\,x_6+\epsilon^7\,x_7$$

Choosing the leading exponent was easy since it was known from the expansion of the exact solution. But — suppose we were tasked with developing an expansion solution for a root of a complicated equation for which we had little insight. One brute force way readily available to a Mathematica user would be to generate a numerical solution and fit it to a polynomial in ϵ

In[46]:=
```
Module[{f, rootNumerical},
 f[x_, ϵ_] := ϵ x^2 + x - 1;
 rootNumerical =
  Table[{ϵ, FindRoot[f[x, ϵ] == 0, {x, -100}][[1, 2]]}, {ϵ, 0.01, 0.1, 0.01}];
 Fit[rootNumerical, {1/ϵ^2, 1/ϵ, 1, ϵ, ϵ^2}, ϵ]]
```

Out[46]=
$$-0.997874 + \frac{1.60693\times 10^{-7}}{\epsilon^2} - \frac{1.00003}{\epsilon} + 0.938648\,\epsilon - 1.17764\,\epsilon^2$$

The smallness of the coefficient of ϵ^{-2} indicates the leading term is ϵ^{-1}. Further on, I'll discuss other methods for determining the expansion rule

In[47]:=
```
w2[4] = w2[1] /. expansionRule
```

Out[47]=
$$-1 + \frac{x_{-1}}{\epsilon} + x_0 + \epsilon\, x_1 + \epsilon^2 x_2 + \epsilon^3 x_3 + \epsilon^4 x_4 + \epsilon^5 x_5 + \epsilon^6 x_6 + \epsilon^7 x_7 + \epsilon\left(\frac{x_{-1}}{\epsilon} + x_0 + \epsilon\, x_1 + \epsilon^2 x_2 + \epsilon^3 x_3 + \epsilon^4 x_4 + \epsilon^5 x_5 + \epsilon^6 x_6 + \epsilon^7 x_7\right)^2 == 0$$

The function CoefficientList used above does not work when there are negative exponents, so I write my own algorithm for extracting the equations.

In[48]:=
```
w2[5] = Module[{expansion, termm1, term0, termp, eqns},
   expansion = ExpandAll[w2[4][[1]] ];
   (* Note the technique used for ϵ^-1 *)
   termm1 = Select[expansion, Not@FreeQ[#, Power[ϵ, -1]] &];
   term0 = Select[expansion, FreeQ[#, ϵ] &];
   termp = CoefficientList[expansion - termm1 - term0, ϵ][[2 ;; -1]];
   eqns = {termm1 /. ϵ → 1, term0, termp} // Flatten;
   (# == 0) & /@ eqns];
w2[5] // ColumnForm
```

Out[49]=

$x_{-1} + x_{-1}^2 == 0$
$-1 + x_0 + 2\, x_{-1}\, x_0 == 0$
$x_0^2 + x_1 + 2\, x_{-1}\, x_1 == 0$
$2\, x_0\, x_1 + x_2 + 2\, x_{-1}\, x_2 == 0$
$x_1^2 + 2\, x_0\, x_2 + x_3 + 2\, x_{-1}\, x_3 == 0$
$2\, x_1\, x_2 + 2\, x_0\, x_3 + x_4 + 2\, x_{-1}\, x_4 == 0$
$x_2^2 + 2\, x_1\, x_3 + 2\, x_0\, x_4 + x_5 + 2\, x_{-1}\, x_5 == 0$
$2\, x_2\, x_3 + 2\, x_1\, x_4 + 2\, x_0\, x_5 + x_6 + 2\, x_{-1}\, x_6 == 0$
$x_3^2 + 2\, x_2\, x_4 + 2\, x_1\, x_5 + 2\, x_0\, x_6 + x_7 + 2\, x_{-1}\, x_7 == 0$
$2\, x_3\, x_4 + 2\, x_2\, x_5 + 2\, x_1\, x_6 + 2\, x_0\, x_7 == 0$
$x_4^2 + 2\, x_3\, x_5 + 2\, x_2\, x_6 + 2\, x_1\, x_7 == 0$
$2\, x_4\, x_5 + 2\, x_3\, x_6 + 2\, x_2\, x_7 == 0$
$x_5^2 + 2\, x_4\, x_6 + 2\, x_3\, x_7 == 0$
$2\, x_5\, x_6 + 2\, x_4\, x_7 == 0$
$x_6^2 + 2\, x_5\, x_7 == 0$
$2\, x_6\, x_7 == 0$
$x_7^2 == 0$

Easily solved

In[50]:=
```
w2[6] = Table[Solve[w2[5][[i]], x_{i-2}][[1, 1]] , {i, 1, 9}]
```

Out[50]=

$$\left\{x_{-1} \to -1,\ x_0 \to \frac{1}{1+2\,x_{-1}},\ x_1 \to -\frac{x_0^2}{1+2\,x_{-1}},\ x_2 \to -\frac{2\,x_0\,x_1}{1+2\,x_{-1}},\right.$$
$$x_3 \to \frac{-x_1^2-2\,x_0\,x_2}{1+2\,x_{-1}},\ x_4 \to -\frac{2\,(x_1\,x_2+x_0\,x_3)}{1+2\,x_{-1}},\ x_5 \to \frac{-x_2^2-2\,x_1\,x_3-2\,x_0\,x_4}{1+2\,x_{-1}},$$
$$\left. x_6 \to -\frac{2\,(x_2\,x_3+x_1\,x_4+x_0\,x_5)}{1+2\,x_{-1}},\ x_7 \to \left(-x_3^2-2\,x_2\,x_4-2\,x_1\,x_5-2\,x_0\,x_6\right) \big/ \left(1+2\,x_{-1}\right)\right\}$$

The explicit coefficients are

In[51]:=
```
w2[7] = Join[{w2[6][[1]]}, Table[w2[6][[i]] //. w2[6][[1 ;; i - 1]], {i, 2, 9}] ]
```

Out[51]=

$\{x_{-1} \to -1,\ x_0 \to -1,\ x_1 \to 1,\ x_2 \to -2,\ x_3 \to 5,\ x_4 \to -14,\ x_5 \to 42,\ x_6 \to -132,\ x_7 \to 429\}$

In[52]:=
```
w2[8] = expansionRule /. w2[7]
```
Out[52]=
$$x \to -1 - \frac{1}{\epsilon} + \epsilon - 2\,\epsilon^2 + 5\,\epsilon^3 - 14\,\epsilon^4 + 42\,\epsilon^5 - 132\,\epsilon^6 + 429\,\epsilon^7$$

Recall the result of the expansion of the exact solution

In[53]:=
```
w2[3]⟦1⟧
```
Out[53]=
$$x \to -1 - \frac{1}{\epsilon} + \epsilon - 2\,\epsilon^2 + 5\,\epsilon^3 - 14\,\epsilon^4 + 42\,\epsilon^5 - 132\,\epsilon^6 + 429\,\epsilon^7$$

which is identical to the series just calculated

In[54]:=
```
w2[8]⟦2⟧ - w2[3]⟦1, 2⟧
```
Out[54]=
```
0
```

Of course, if one found oneself performing repetitive calculations of this sort, it would be easy enough to collect the required operations into a function that could called for each new calculation.

3 A problem involving expansion in fractional powers of ϵ

In Section 1.3, Hinch discusses a example that requires an expansion in terms of fractional power of ϵ

In[56]:=
```
w3[1] = (1 - ϵ) x^2 - 2 x + 1 == 0
```
Out[56]=
$$1 - 2\,x + x^2\,(1-\epsilon) == 0$$

I first take advantage of the available exact solution to generate the series that will be later calculated

In[57]:=
```
w3[2] = Solve[w3[1], x]
```
Out[57]=
$$\left\{\left\{x \to \frac{-1-\sqrt{\epsilon}}{-1+\epsilon}\right\}, \left\{x \to \frac{-1+\sqrt{\epsilon}}{-1+\epsilon}\right\}\right\}$$

In[58]:=
```
w3[3] = w3[2] /. ϵ → 0
```
Out[58]=
```
{{x → 1}, {x → 1}}
```

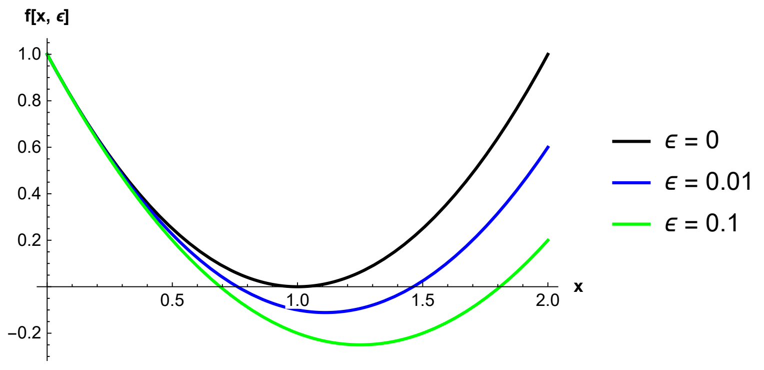
In this case, the expansion involves fractional powers.

In[59]:=
```
w3[3] = (#⟦1, 1⟧ → Normal@Series[#⟦1, 2⟧, {ϵ, 0, 3}]) & /@ w3[2]
```
Out[59]=
$$\left\{x \to 1 + \sqrt{\epsilon} + \epsilon + \epsilon^{3/2} + \epsilon^2 + \epsilon^{5/2} + \epsilon^3,\ x \to 1 - \sqrt{\epsilon} + \epsilon - \epsilon^{3/2} + \epsilon^2 - \epsilon^{5/2} + \epsilon^3\right\}$$

A visualization provides some insight — there are double roots at x = 1 for ϵ = 0, but these roots separate for finite ϵ

In[60]:=
```
Module[{f},
 f[x_, ϵ_] := (1 - ϵ) x^2 - 2 x + 1;
 Plot[{f[x, 0], f[x, 0.1], f[x, 0.2]}, {x, 0, 2},
  PlotStyle → {Black, Blue, Green}, AxesLabel → {Stl["x"], Stl["f[x, ϵ]"]},
  PlotLegends → {"ϵ = 0", "ϵ = 0.01", "ϵ = 0.1"}, ImageSize → 300]]
```

Out[60]=



One obstacle when using the expansion method is that a reasonable guess must be made for the form of the expansion. In this case the explicit form is known from the exact solution. But, suppose that we don’t have an exact solution. I give two methods for guessing the form of the expansion.

3a Visualize the roots on a log log plot

Numerically determine the roots and plot the deviation from their $\epsilon = 0$ value. In the following I consider the branch for which the roots are less than 1 as ϵ increases, and plot log[1 - root] as a function of log[ϵ] for small ϵ.

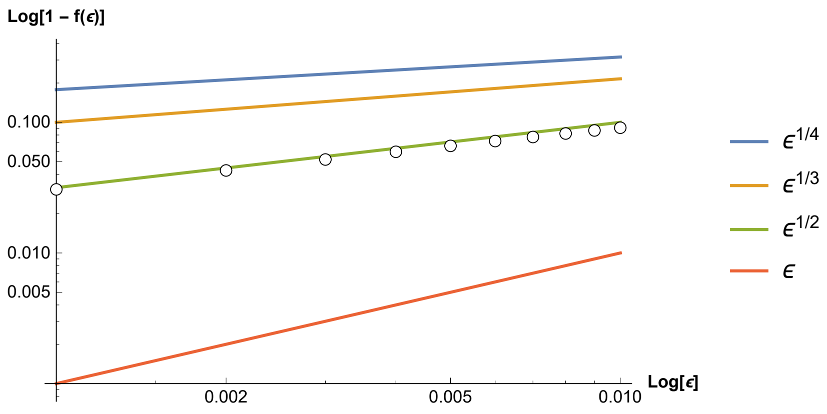
In[61]:=

```
Module[{f, rootGuess, roots, rootDiffs, points},
 f[x_, ϵ_] := (1 - ϵ) x² - 2 x + 1;

 rootGuess = 0.5;
 roots =
  Table[{ϵ, FindRoot[f[x, ϵ] , {x, rootGuess}][[1, 2]]}, {ϵ, 0.001, 0.01, 0.001}];
 rootDiffs = ({Log[#[[1]]], Log[1 - #[[2]]]}) & /@ roots;
 points = OC[#, Black] & /@ rootDiffs;

 LogLogPlot[{ϵ^1/4, ϵ^1/3, ϵ^1/2, ϵ}, {ϵ, 0.001, 0.01},
  Epilog → points, PlotLegends → {"ϵ^1/4", "ϵ^1/3", "ϵ^1/2", "ϵ"},
  AxesLabel → {Stl["Log[ϵ]"], Stl["Log[1 - f(ϵ)]"]}]]
```

Out[61]=



From this figure I surmise the expansion should be

$$a_0 + a_{\frac{1}{2}}\,\epsilon^{\frac{1}{2}} + a_1\,\epsilon + \ldots$$

A beginning student might reasonably ask the question - Why expend so much effort developing a perturbation solution when a numerical solution is easily obtained?" The answer is that analytical solutions are **extraordinarily** useful in science and mathematics. They allow one to reason about mechanisms, to generalize to other problems, to postpone the use of numerical methods to later stages of analysis, and much more. As Richard Hamming said, "the purpose of computing is insight, not numbers."

3b Fit the deviation of roots from their $\epsilon = 0$ to a functional form

Numerically determine the roots and plot the deviation from their $\epsilon = 0$ value. In the following I consider the branch for which the roots are less than 1 as ϵ increases. I assume that 1 - root(ϵ) = a ϵ^b and use the function NonlinearModelFit to estimate a and b

In[90]:=
```
Module[{f, rootGuess, roots, rootDiffs, points},
 f[x_, ϵ_] := (1 - ϵ) x^2 - 2 x + 1;

 rootGuess = 0.5;
 roots =
  Table[{ϵ, FindRoot[f[x, ϵ] , {x, rootGuess}][[1, 2]]}, {ϵ, 0.001, 0.01, 0.001}];
 rootDiffs = ({#[[1]], 1 - #[[2]]}) & /@ roots;

 NonlinearModelFit[rootDiffs, a ϵ^b, {a, b}, ϵ]]
```

Out[90]= FittedModel[$0.788439\, \epsilon^{0.468529}$]

From the fitted result a = 0.788, b = 0.468 I again surmise the expansion should be

$$a_0 + a_{\frac{1}{2}}\, \epsilon^{\frac{1}{2}} + a_1\, \epsilon + \ldots$$

3c Application of expansion method

Armed with a good idea of an appropriate form of the expansion I assume

In[63]:=
```
expansionRule = x → 1 + Sum[ϵ^(i/2) x_(i/2), {i, 1, 4}]
```

Out[63]= $x \to 1 + \sqrt{\epsilon}\, x_{\frac{1}{2}} + \epsilon\, x_1 + \epsilon^{3/2}\, x_{\frac{3}{2}} + \epsilon^2\, x_2$

Then

In[64]:=
```
w3c[1] = w3[1] /. expansionRule
```

Out[64]= $1 - 2\left(1 + \sqrt{\epsilon}\, x_{\frac{1}{2}} + \epsilon\, x_1 + \epsilon^{3/2}\, x_{\frac{3}{2}} + \epsilon^2\, x_2\right) + (1 - \epsilon)\left(1 + \sqrt{\epsilon}\, x_{\frac{1}{2}} + \epsilon\, x_1 + \epsilon^{3/2}\, x_{\frac{3}{2}} + \epsilon^2\, x_2\right)^2 == 0$

This leads to the equations

In[65]:=
```
w3c[2] = Module[{expansion, eqns},
   (*CoefficientList can be used if
    the expansion variable is forced to have integer powers*)
   expansion = ExpandAll[w3c[1][[1]]] /. ϵ → η^2 // PowerExpand;
   eqns = CoefficientList[expansion, η];
   (# == 0) & /@ eqns];
w3c[2] // ColumnForm
```

Out[66]=
```
True
True
```
$$-1 + x_{\frac{1}{2}}^2 == 0$$
$$-2\, x_{\frac{1}{2}} + 2\, x_{\frac{1}{2}}\, x_1 == 0$$
$$-x_{\frac{1}{2}}^2 - 2\, x_1 + x_1^2 + 2\, x_{\frac{1}{2}}\, x_{\frac{3}{2}} == 0$$
$$-2\, x_{\frac{1}{2}}\, x_1 - 2\, x_{\frac{3}{2}} + 2\, x_1\, x_{\frac{3}{2}} + 2\, x_{\frac{1}{2}}\, x_2 == 0$$
$$-x_1^2 - 2\, x_{\frac{1}{2}}\, x_{\frac{3}{2}} + x_{\frac{3}{2}}^2 - 2\, x_2 + 2\, x_1\, x_2 == 0$$
$$-2\, x_1\, x_{\frac{3}{2}} - 2\, x_{\frac{1}{2}}\, x_2 + 2\, x_{\frac{3}{2}}\, x_2 == 0$$
$$-x_{\frac{3}{2}}^2 - 2\, x_1\, x_2 + x_2^2 == 0$$
$$-2\, x_{\frac{3}{2}}\, x_2 == 0$$
$$-x_2^2 == 0$$

The two lowest order equations in ϵ^0 and $\epsilon^{\frac{1}{2}}$ provide no information

The equation in ϵ provides two values for $x_{\frac{1}{2}}$!

In[67]:=
```
w3c[3] = Solve[w3c[2][[3]], x_{1/2}]
```

Out[67]=
$$\left\{\left\{x_{\frac{1}{2}} \to -1\right\}, \left\{x_{\frac{1}{2}} \to 1\right\}\right\}$$

No multiplicity at order $\epsilon^{\frac{3}{2}}$

In[68]:=
```
w3c[4] = Solve[w3c[2][[4]], x_1 ][[1, 1]]
```

Out[68]=
$$x_1 \to 1$$

Proceeding

In[69]:=
```
w3c[5] = Solve[w3c[2][[5]], x_{3/2} ][[1, 1]]
```

Out[69]=
$$x_{\frac{3}{2}} \to \frac{x_{\frac{1}{2}}^2 + 2\, x_1 - x_1^2}{2\, x_{\frac{1}{2}}}$$

In[70]:=
```
w3c[6] = Solve[w3c[2][[6]], x₂ ][[1, 1]]
```
Out[70]=
$$x_2 \to \frac{x_{\frac{1}{2}}\, x_1 + x_{\frac{3}{2}} - x_1\, x_{\frac{3}{2}}}{x_{\frac{1}{2}}}$$

The general expansion is

In[71]:=
```
w3c[7] = expansionRule /. w3c[6] /. w3c[5] /. w3c[4]
```
Out[71]=
$$x \to 1 + \epsilon + \epsilon^2 + \sqrt{\epsilon}\, x_{\frac{1}{2}} + \frac{\epsilon^{3/2}\left(1 + x_{\frac{1}{2}}^2\right)}{2\, x_{\frac{1}{2}}}$$

and the specific forms are

In[72]:=
```
w3c[8] = {w3c[7] /. w3c[3][[1]], w3c[7] /. w3c[3][[2]]}
```
Out[72]=
$$\left\{x \to 1 - \sqrt{\epsilon} + \epsilon - \epsilon^{3/2} + \epsilon^2,\ x \to 1 + \sqrt{\epsilon} + \epsilon + \epsilon^{3/2} + \epsilon^2\right\}$$

in agreement with the expansion of the exact solutions.

I have explicitly shown all the steps in this example. The various sequential operations could be collected into a single function if there were sufficient motivation.

4 Another method for determining an appropriate expansion

Departing from examples in Hinch I turn to Section 2.2 of *Introduction to the Foundations of Applied Mathematics*, Mark H. Holmes. Holmes discusses a general method for determining expansions and applies it to different example problems. I implement this method in Mathematica for a problem similar to those treated in previous sections.

Consider the regular perturbation problem

In[91]:=
```
w4[1] = x^2 + 2 ϵ x - 1 == 0
```
Out[91]=
$$-1 + x^2 + 2\, x\, \epsilon == 0$$

Holmes suggests the general expansion form

$$x \to x_0 + x_1\, \epsilon^{\alpha} + x_2\, \epsilon^{\beta} + x_3\, \epsilon^{\gamma} + \ldots$$

where it is assumed $\alpha << \beta << \gamma << \ldots$ so that $\epsilon^{\alpha} >> \epsilon^{\beta} >> \epsilon^{\gamma} >> \ldots$

I will consider only leading terms by explicitly choosing

```
In[92]:= expansionRule = x → x0 + ϵ^α x1 + ϵ^β x2
```

Out[92]= $x \to x_0 + \epsilon^{\alpha} x_1 + \epsilon^{\beta} x_2$

```
In[93]:= w4[2] = w4[1] /. expansionRule // ExpandAll
```

Out[93]= $-1 + 2 \epsilon x_0 + x_0^2 + 2 \epsilon^{1+\alpha} x_1 + 2 \epsilon^{\alpha} x_0 x_1 + \epsilon^{2\alpha} x_1^2 + 2 \epsilon^{1+\beta} x_2 + 2 \epsilon^{\beta} x_0 x_2 + 2 \epsilon^{\alpha+\beta} x_1 x_2 + \epsilon^{2\beta} x_2^2 == 0$

At lowest order

```
In[94]:= w4[3] = w4[2] /. ϵ → 0 /. 0- → 0
```

Out[94]= $-1 + x_0^2 == 0$

there are two roots

```
In[95]:= w4[4] = Solve[w4[3], x0]
```

Out[95]= $\{\{x_0 \to -1\}, \{x_0 \to 1\}\}$

Normally I would substitute one of these values for x_0 into the perturbation expression and calculate the next order correction. In this case I just substitute the zero order equation and express the higher order corrections in terms of x_0.

```
In[96]:= w4[5] = w4[2] /. x0^2 → 1
```

Out[96]= $2 \epsilon x_0 + 2 \epsilon^{1+\alpha} x_1 + 2 \epsilon^{\alpha} x_0 x_1 + \epsilon^{2\alpha} x_1^2 + 2 \epsilon^{1+\beta} x_2 + 2 \epsilon^{\beta} x_0 x_2 + 2 \epsilon^{\alpha+\beta} x_1 x_2 + \epsilon^{2\beta} x_2^2 == 0$

At this stage, the lowest order term that remains is linear in ϵ, the plan is to determine the value of α that will allow that term to be balanced.

I note that the term linear in ϵ can be selected from the general expression with the operation

```
In[97]:= Select[w4[5][[1]] , (MatchQ[#, a_. ϵ]) &]
```

Out[97]= $2 \epsilon x_0$

and those terms containing ϵ^{α} with

```
In[98]:= Select[w4[5][[1]] , (Not@FreeQ[#, ϵ^α]) &]
```

Out[98]= $2 \epsilon^{\alpha} x_0 x_1$

or with

```
In[99]:=  Select[w4[5][[1]], (MatchQ[#, a_. ϵ^α]) &]

Out[99]=  2 ϵ^α x0 x1
```

Thus

```
In[100]:= w4[6] = Select[w4[5][[1]], (Or[MatchQ[#, a_. ϵ], MatchQ[#, a_. ϵ^α]]) &] == 0

Out[100]= 2 ϵ x0 + 2 ϵ^α x0 x1 == 0
```

All other terms, such as $\epsilon^{\alpha+1}$ or ϵ^{β} will be much smaller

From this equation, both α and x_1 must be determined. I'll illustrate an algorithm that accomplishes these tasks. I first proceed step by step and then encapsulate the operations into a general function

An equation for α can be obtained with

```
In[101]:= aside[1] = (w4[6][[1]] /. a_ ϵ + b_ ϵ^α → ϵ - ϵ^α) == 0

Out[101]= ϵ - ϵ^α == 0
```

and solved

```
In[102]:= aside[2] = Quiet@Solve[aside[1], α][[1, 1]]

Out[102]= α → 1
```

Then,

```
In[103]:= aside[3] = {aside[2], Solve[w4[6] /. aside[2], x1][[1, 1]]}

Out[103]= {α → 1, x1 → -1}
```

and the order ϵ correction has been obtained.

I define the function

```
In[104]:= Clear[BalanceAtOrder];
          BalanceAtOrder[eqn_, ϵOrder_, orderVariable_, perturbationVariable_] :=
           Module[{orderVariableValue, perturbationVariableValue},
            orderVariableValue = Quiet@Solve[ϵOrder - ϵ^orderVariable == 0, orderVariable][[1, 1]];
            perturbationVariableValue =
             Solve[(eqn /. orderVariableValue), perturbationVariable][[1, 1]];
            {orderVariableValue, perturbationVariableValue} ]
```

Using this function, the order ϵ balancing terms are obtain immediately

```
In[106]:= w4[7] = BalanceAtOrder[w4[6], ϵ, α, x1]

Out[106]= {α → 1, x1 → -1}
```

Proceeding to the next order

```
In[110]:= w4[8] = w4[5] /. w4[7]
```

Out[110]= $-\epsilon^2 + 2\,\epsilon^\beta\, x_0\, x_2 + \epsilon^{2\beta}\, x_2^2 == 0$

Select the terms that must balance

```
In[113]:= w4[9] = Select[w4[8][[1]], (Or[MatchQ[#, a_. ϵ^2], MatchQ[#, a_. ϵ^β]]) &] == 0
```

Out[113]= $-\epsilon^2 + 2\,\epsilon^\beta\, x_0\, x_2 == 0$

Then

```
In[114]:= w4[10] = BalanceAtOrder[w4[9], ϵ^2, β, x2] // PowerExpand
```

Out[114]= $\left\{\beta \to 2,\ x_2 \to \frac{1}{2\, x_0}\right\}$

So, the expansionRule can be made specific

```
In[115]:= w4[11] = expansionRule /. w4[7] /. w4[10]
```

Out[115]= $x \to -\epsilon + \frac{\epsilon^2}{2\, x_0} + x_0$

I compare this result with the solution of the original equation. The original equation was quadratic and so the exact solution is available. But — suppose the exact solution was not available. I illustrate a numerical approach.

```
In[116]:= NUMERICALSOLUTION = Flatten /@ Table[{ϵ, x /. NSolve[w4[1], x]}, {ϵ, 0, 1, 0.1}]
```

```
Out[116]= {{0., -1., 1.}, {0.1, -1.10499, 0.904988}, {0.2, -1.2198, 0.819804},
  {0.3, -1.34403, 0.744031}, {0.4, -1.47703, 0.677033},
  {0.5, -1.61803, 0.618034}, {0.6, -1.76619, 0.56619}, {0.7, -1.92066, 0.520656},
  {0.8, -2.08062, 0.480625}, {0.9, -2.24536, 0.445362}, {1., -2.41421, 0.414214}}
```

where, in this case the equation is a polynomial and NSolve can be used. If the equation was transcendental, I would use FindRoot.

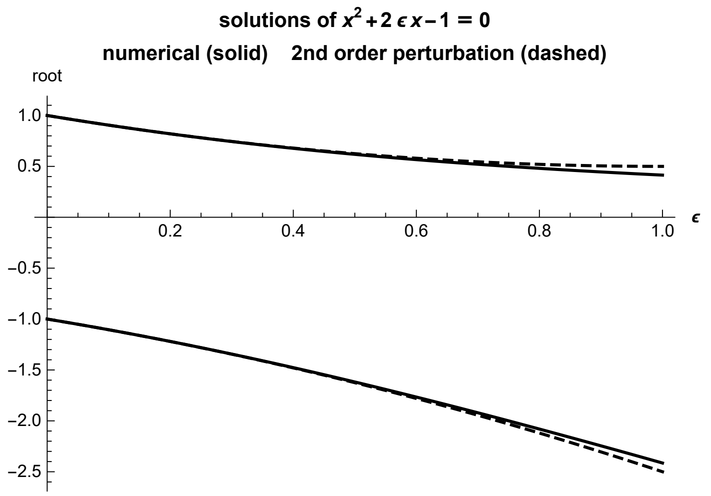
The perturbation solution is

```
In[117]:= Clear[RootPerturbation];
RootPerturbation[ϵ_, x0_] := -ϵ + ϵ^2/(2 x0) + x0
```

Then

In[119]:=
```
Module[{root, ϵMax, x0, lab},
 lab = Stl@StringForm[
    "solutions of ``\nnumerical (solid)    2nd order perturbation (dashed)",
    TraditionalForm[w4[1]]];
 ϵMax = NUMERICALSOLUTION[[-1, 1]];
 root[1] = Interpolation@NUMERICALSOLUTION[[All, {1, 2}]];
 root[2] = Interpolation@NUMERICALSOLUTION[[All, {1, 3}]];
 x0[1] = 1; x0[2] = -1;
 Plot[{root[1][ϵ], RootPerturbation[ϵ, x0[1]],
   root[2][ϵ], RootPerturbation[ϵ, x0[2]]}, {ϵ, 0, ϵMax},
  PlotStyle → {Black, Directive[Black, Dashed]}, AxesLabel → {Stl[ϵ], "root"},
  PlotLabel → lab]]
```

Out[119]=



Agreement is quite good, even for ϵ of order 1.

5 A problem for which iteration is preferred

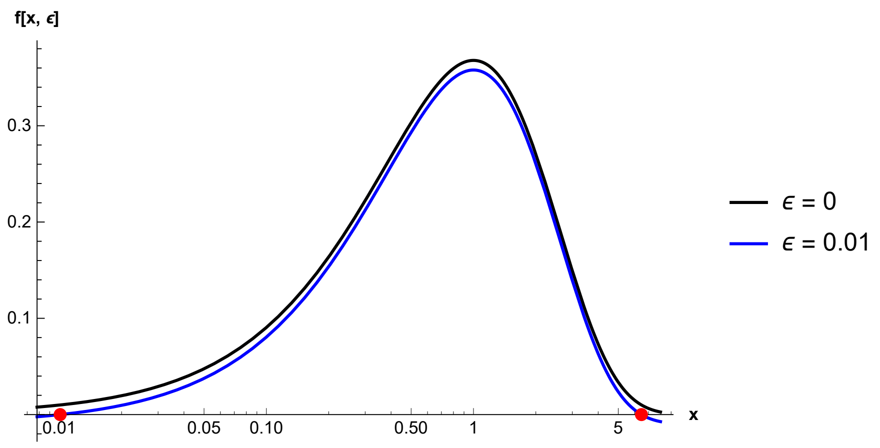
Hinch 1.4 considers the transcendental equation

In[120]:=
```
w5[1] = x Exp[-x] == ϵ
```

Out[120]=
```
e^-x x == ϵ
```

I visualize the location of roots using

In[121]:=
```
Module[{f, root, line, points},
 f[x_, ϵ_] := x Exp[-x] - ϵ;
 root[1] = With[{xGuess = 0.01}, FindRoot[f[x, 0.01] == 0, {x, xGuess}]][[1, 2]] ;
 root[2] = With[{xGuess = 5}, FindRoot[f[x, 0.01] == 0, {x, xGuess}]][[1, 2]];
 points =
  {PointSize[0.02], Red, Point[{Log@root[1], 0}], Point[{Log@root[2], 0}]};
 line = {Directive[Black, Dashed], Line[{{-Log[0.01], 0}, {-Log[0.01], 1}}] };
 LogLinearPlot[{f[x, 0], f[x, 0.01]}, {x, 0, 8},
  PlotStyle → {Black, Blue}, Epilog → {line, points},
  AxesLabel → {Stl["x"], Stl["f[x, ϵ]"]}, PlotLegends → {"ϵ = 0", "ϵ = 0.01"} ]]
```

Out[121]=


Hinch suggests an iteration scheme appropriate for the root at large x

In[122]:=
```
w5[2] = MapEqn[(# / x) &, w5[1]]
```

Out[122]=
$$e^{-x} == \frac{\epsilon}{x}$$

In[123]:=
```
w5[3] = MapEqn[(-Log[#]) &, w5[2]] // PowerExpand
```

Out[123]=
$$x == \mathrm{Log}[x] - \mathrm{Log}[\epsilon]$$

In[124]:=
```
w5[4] = (w5[3][[1]] /. x → x_{n+1}) == (w5[3][[2]] /. x → x_n)
```

Out[124]=
$$x_{1+n} == -\mathrm{Log}[\epsilon] + \mathrm{Log}[x_n]$$

The lowest order approximation is

In[125]:=
```
w5[5] = x_0 == -Log[ϵ]
```

Out[125]=
$$x_0 == -\mathrm{Log}[\epsilon]$$

To first order

In[126]:=
```
w5[6] = w5[4] /. n → 0 /. (w5[5] // ER)
```

Out[126]= $x_1 == -\text{Log}[\epsilon] + \text{Log}[-\text{Log}[\epsilon]]$

and 2rd order

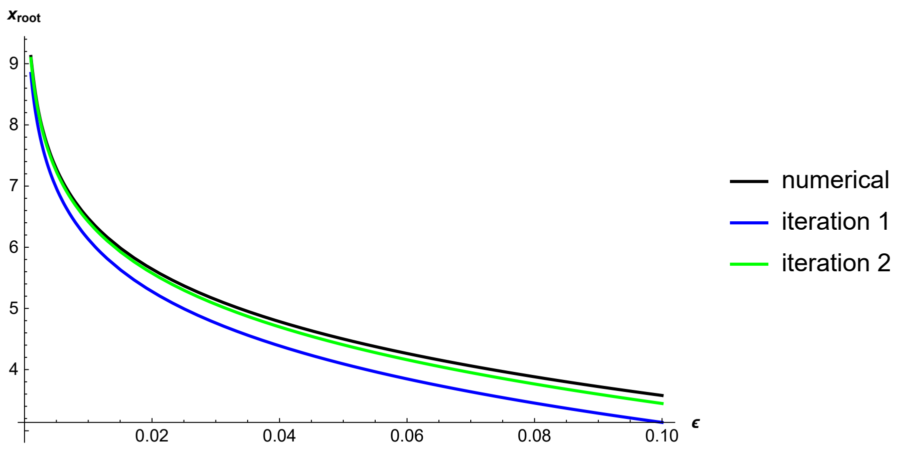
In[127]:=
```
w5[7] = w5[4] /. n → 1 /. (w5[6] // ER)
```

Out[127]= $x_2 == -\text{Log}[\epsilon] + \text{Log}[-\text{Log}[\epsilon] + \text{Log}[-\text{Log}[\epsilon]]]$

Hinch makes the point that this would not have been an easy expansion to guess.

In[128]:=
```
Clear[RootOrder1, RootOrder2];
RootOrder1[ϵ_] := -Log[ϵ] + Log[-Log[ϵ]];
RootOrder2[ϵ_] := -Log[ϵ] + Log[-Log[ϵ] + Log[-Log[ϵ]]];
```

In[131]:=
```
Module[{f, fRoot, rootNumerical},
 f[x_, ϵ_] := x Exp[-x] - ϵ;
 fRoot[ϵ_] := FindRoot[f[x, ϵ] == 0, {x, -Log[ϵ]}][[1, 2]];
 rootNumerical = Interpolation@Table[{ϵ, fRoot[ϵ]}, {ϵ, 0.1, 0.001, -0.001}];
 Plot[{rootNumerical[ϵ], RootOrder1[ϵ], RootOrder2[ϵ]},
  {ϵ, 0.001, 0.1}, PlotStyle → {Black, Blue, Green},
  PlotLegends → {"numerical", "iteration 1", "iteration 2"},
  AxesLabel → {Stl["ϵ"], Stl["x_root"] }]]
```

Out[131]=


Hinch also makes the point that quite small values of ϵ are required to achieve high accuracy