

Nov. 7 2007

## Lecture 20: Linear Homogeneous and Heterogeneous ODEs

Reading:

Kreyszig Sections: 1.4, 1.5 (pages 19–25, 26–32)

### Ordinary Differential Equations from Physical Models

In engineering and physics, modeling physical phenomena is the means by which technological and natural phenomena are understood and predicted. A model is an abstraction of a physical system, often with simplifying assumptions, into a mathematical framework. Every model should be verifiable by an experiment that, to the greatest extent possible, satisfies the approximations that were used to obtain the model.

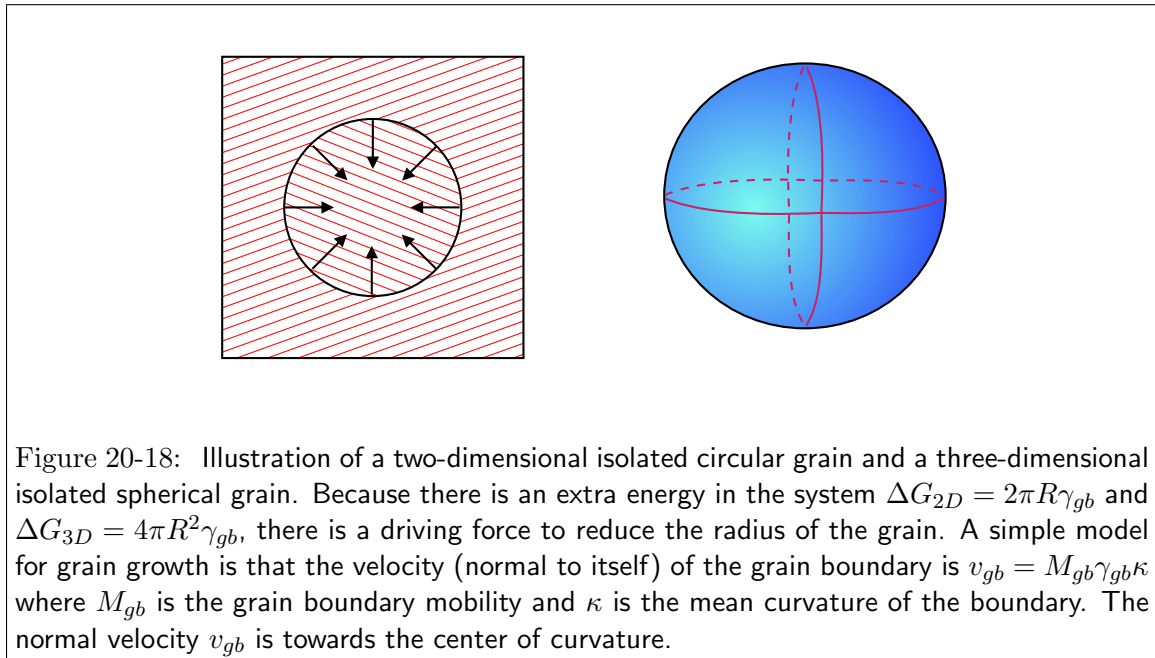
In the context of modeling, differential equations appear frequently. Learning how to model new and interesting systems is a learned skill—it is best to learn by following a few examples. Grain growth provides some interesting modeling examples that result in first-order ODEs.

### Grain Growth

In materials science and engineering, a grain usually refers a single element in an ensemble that comprises a polycrystal. In a single phase polycrystal, a grain is a contiguous region of material with the same crystallographic orientation. It is separated from other grains by *grain boundaries* where the crystallographic orientation changes abruptly.

A grain boundary contributes extra free energy to the entire system that is proportional to the grain boundary area. Thus, if the boundary can move to reduce the free energy it will.

Consider simple, uniformly curved, isolated two- and three-dimensional grains.



A relevant question is “how fast will a grain change its size assuming that grain boundary migration velocity is proportional to curvature?”

For the two-dimensional case, the rate of change of area can be formulated by considering the following illustration.

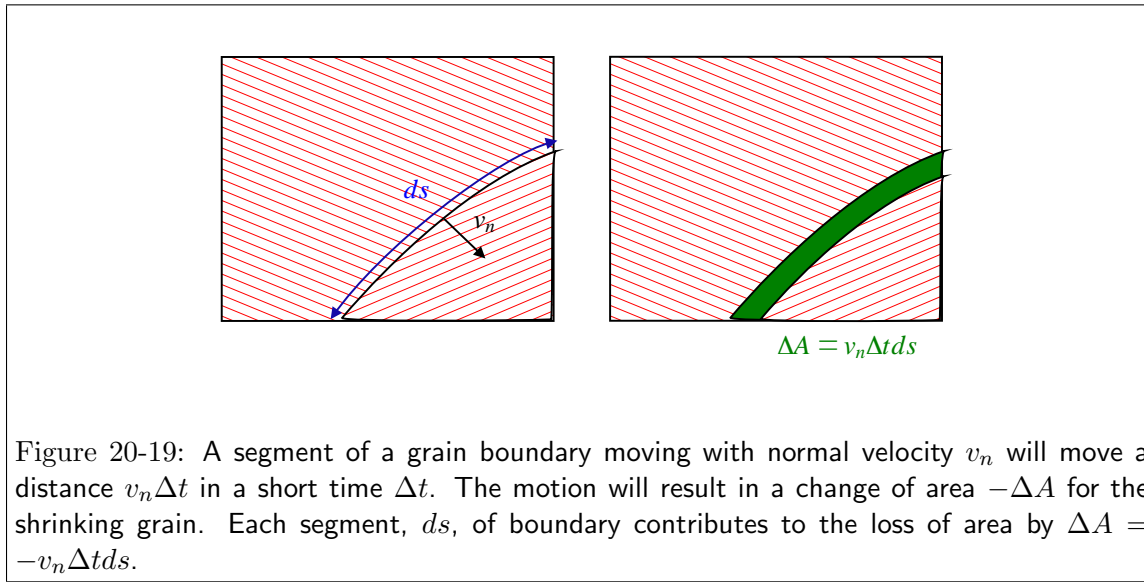


Figure 20-19: A segment of a grain boundary moving with normal velocity  $v_n$  will move a distance  $v_n \Delta t$  in a short time  $\Delta t$ . The motion will result in a change of area  $-\Delta A$  for the shrinking grain. Each segment,  $ds$ , of boundary contributes to the loss of area by  $\Delta A = -v_n \Delta t ds$ .

Because for a circle, the curvature is the same at each location on the grain boundary, the curvature is uniform and  $v_n = M_{gb} \kappa_{gb} \gamma_{gb} = M_{gb} \gamma_{gb} / R$ . Thus

$$\frac{dA}{dt} = -M_{gb} \gamma_{gb} \frac{1}{R} 2\pi R = -2\pi M_{gb} \gamma_{gb} \quad (20-1)$$

Thus, the area of a circular grain changes at a constant rate, the rate of change of radius is:

$$\frac{dA}{dt} = \frac{d\pi R^2}{dt} = 2\pi R \frac{dR}{dt} = -2\pi M_{gb} \gamma_{gb} \quad (20-2)$$

which is a first-order, separable ODE with solution:

$$R^2(t) - R^2(t=0) = -2M_{gb} \gamma_{gb} t \quad (20-3)$$

For a spherical grain, the change in volume  $\Delta V$  due to the motion of a surface patch  $dS$  in a time  $\Delta t$  is  $\Delta V = v_n \Delta t dS$ . The curvature of a sphere is

$$\kappa_{sphere} = \left( \frac{1}{R} + \frac{1}{R} \right) \quad (20-4)$$

Therefore the velocity of the interface is  $v_n = 2M_{gb} \gamma_{gb} / R$ . The rate of change of volume due to the contributions of each surface patch is

$$\frac{dV}{dt} = -M_{gb} \gamma_{gb} \frac{2}{R} 4\pi R^2 = -8\pi M_{gb} \gamma_{gb} R = -4(6\pi^2)^{1/3} M_{gb} \gamma_{gb} V^{1/3} \quad (20-5)$$

which can be separated and integrated:

$$V^{2/3}(t) - V^{2/3}(t=0) = -\text{constant}_1 t \quad (20-6)$$

or

$$R^2(t) - R^2(t=0) = -\text{constant}_2 t \quad (20-7)$$

which is the same functional form as derived for two-dimensions.

The problem (and result) is more interesting if the grain doesn't have uniform curvature.



$U(S, V) = \text{const}$ , then the above equation becomes a *differential form*

$$0 = \left( \frac{\partial U}{\partial S} \right)_V dS + \left( \frac{\partial U}{\partial V} \right)_S dV \quad (20-9)$$

This equation expresses a relation between changes in  $S$  and changes in  $V$  that are necessary to remain on the surface  $U(S, V) = \text{const}$ .

Suppose the situation is turned around and you are given the first-order ODE

$$\frac{dy}{dx} = -\frac{M(x, y)}{N(x, y)} \quad (20-10)$$

which can be written as the differential form

$$0 = M(x, y)dx + N(x, y)dy \quad (20-11)$$

Is there a function  $U(x, y) = \text{const}$  or, equivalently, is it possible to find a curve represented by  $U(x, y) = \text{const}$ ?

If such a curve exists then it depends only on one parameter, such as arc-length, and on that curve  $dU(x, y) = 0$ .

The answer is, “Yes, such a function  $U(x, y) = \text{const}$  exists if and only if  $M(x, y)$  and  $N(x, y)$  satisfy the Maxwell relations”

$$\frac{\partial M(x, y)}{\partial y} = \frac{\partial N(x, y)}{\partial x} \quad (20-12)$$

Then if Eq. 20-12 holds, the differential form Eq. 20-11 is called an *exact differential* and a  $U$  exists such that  $dU = 0 = M(x, y)dx + N(x, y)dy$ .

---



---



---



---

### Integrating Factors and Thermodynamics

For fixed number of moles of ideal gas, the internal energy is a function of the temperature only,  $U(T) - U(T_o) = C_V(T - T_o)$ . Consider the heat that is transferred to a gas that changes its temperature and volume a very small amount:

$$\begin{aligned} dU &= C_V dT = \delta q + \delta w = \delta q - PdV \\ \delta q &= C_V dT + PdV \end{aligned} \quad (20-13)$$

Can a Heat Function  $q(T, V) = \text{constant}$  be found?

To answer this, apply Maxwell's relations.



Dividing each side through by through by  $y$  and integrate:

$$\int \frac{dy}{y} = \log y = - \int p(x)dx + \text{const}$$

which has solution

$$y(x) = \text{const} e^{-\int p(x)dx}$$

For the case of the heterogeneous first-order ODE, A trick (or, an integrating factor which amounts to the same thing) can be employed. Multiply both sides of the heterogeneous equation by  $e^{\int p(x)}$ .<sup>12</sup>

$$\exp \left[ \int_a^x p(z)dz \right] \frac{dy(x)}{dx} + \exp \left[ \int_a^x p(z)dz \right] p(x)y(x) = \exp \left[ \int_a^x p(z)dz \right] r(x) \quad (20-17)$$

Notice that the left-hand-side can be written as a derivative of a simple expression

$$\exp \left[ \int_a^x p(z)dz \right] \frac{dy(x)}{dx} + \exp \left[ \int_a^x p(z)dz \right] p(x)y(x) = \frac{d}{dx} \left\{ \exp \left[ \int_a^x p(z)dz \right] y(x) \right\} \quad (20-18)$$

therefore

$$\frac{d}{dx} \left\{ \exp \left[ \int_a^x p(z)dz \right] y(x) \right\} = \exp \left[ \int_a^x p(z)dz \right] r(x) \quad (20-19)$$

which can be integrated and then solved for  $y(x)$ :

$$y(x) = \exp \left[ - \int_a^x p(z)dz \right] \left\{ y(x=a) + \int_a^x r(z) \exp \left[ \int_a^z p(\eta)d\eta \right] dz \right\} \quad (20-20)$$

---

<sup>12</sup> The statistical definition of entropy is  $S(T, V) = k \log \Omega(U(T, V))$  or  $\Omega(U(T, V)) = \exp(S/k)$ . Entropy plays the role of integrating factor.

## Lecture 20 MATHEMATICA® Example 1

## Solutions to the General Homogeneous Linear First-Order ODE

Download [notebooks](http://pruffle.mit.edu/3.016-2007), [pdf\(color\)](#), [pdf\(bw\)](#), or [html](#) from <http://pruffle.mit.edu/3.016-2007>.

The form of MATHEMATICA®'s solution for Eq. 20 is demonstrated.

```
DSolve[
  y'[x] + p[x] y[x] == 0, 1
  y[x], x]
```

```
{{y[x] ->
  e^{\int -p[k[1]] dk[1]} c[1]}}
```

*The dummy integration variables (K[1] in the above) and any integration constants (C[1] above) are picked by Mathematica. Mathematica returns the most general form of homogeneous linear first-order solution.*

```
DSolve[
  y'[x] + (2 x + 1) y[x] == 2
  0, y[x], x] 2
```

```
{{y[x] -> e^{-x-x^2} c[1]}}
```

*There is an integration constant above, that will take on a specific value if an additional condition (such as an initial condition, or a boundary condition) is specified*

```
DSolve[{y'[x] +
  (2 x + 1) y[x] == 0,
  y[0] == 4}, y[x], x] 3
```

```
{{y[x] -> 4 e^{-x-x^2}}}
```

- 1: DSolve solves the linear homogeneous equation first-order ODE  $dy/dx + p(x)y = 0$ . Two variables are introduced in the solution: one is the 'dummy-variable' of the integration in Eq. 20 which MATHEMATICA® introduces in the form  $K[N]$  and an integration constant which is given the form  $C[N]$ .
- 2: Here, a specific  $p(x)$  is given, so the dummy variable doesn't appear if  $p(\zeta)$  can be integrated symbolically, as in this case for  $p(\zeta) = 2x + 1$ .
- 3: Furthermore, if enough boundary conditions are given to solve for the integration constants, then the  $C[N]$  are not needed either.

## Lecture 20 MATHEMATICA® Example 2

## Solutions to the General Heterogeneous Linear First-Order ODE

Download [notebooks](#), [pdf\(color\)](#), [pdf\(bw\)](#), or [html](http://pruffle.mit.edu/3.016-2007) from <http://pruffle.mit.edu/3.016-2007>.

This demonstrates the use of `DSolve` to find symbolic solutions of heterogeneous linear homogeneous first-order ODEs: Eq. 20-20. We will see how the homogeneous solution is always part of the sum for a heterogeneous solution.

```

1
DSolve[
2
  y'[x] + p[x] y[x] ==
  r[x], y[x], x]
3

```

$$\left\{ \left\{ y[x] \rightarrow \begin{aligned} &e^{\int_0^x -p[K[1]] dK[1]} C[1] + \\ &e^{\int_0^x -p[K[1]] dK[1]} \int_1^x e^{-\int_1^z -p[K[1]] dK[1]} r[K[2]] dK[2] \end{aligned} \right\} \right\}$$

*The solution is general—two dummy integration variables and one constant of integration.*

```

homsol = DSolve[
  y'[x] - y[x] == 0,
  y[x], x]
hetsol = DSolve[
  y'[x] - y[x] == e2x,
  y[x], x]

```

```

{{y[x] -> ex C[1]}}
{{y[x] -> e2x + ex C[1]}}

```

- 1: `DSolve` solves the general linear heterogeneous equation,  $dy/dx + p(x)y = r(x)$ , to give the form Eq. 20-20. Note how the homogeneous solution (i.e., the part that depends on `C[1]`) is part of the solution.
- 2: This is an example for a specific case:  $p(x) = -1$  and  $r(x) = e^{2x}$ . The homogeneous solution is displayed alongside to reinforce that it is always part of the solution.

## Example: The Bernoulli Equation

The linear first-order ODEs always have a closed form solution in terms of integrals. In general non-linear ODEs do not have a general expression for their solution. However, there are some non-linear equations that can be reduced to a linear form; one such case is the Bernoulli equation:

$$\frac{dy}{dx} + p(x)y = r(x)y^a \quad (20-21)$$

Reduction relies on a clever change-of-variable, let  $u(x) = [y(x)]^{1-a}$ , then Eq. 20-21 becomes

$$\frac{du}{dx} + (1-a)p(x)u = (1-a)r(x) \quad (20-22)$$



which is a linear heterogeneous first-order ODE and has a closed-form solution.

However, not all non-linear problems can be converted to a linear form. In these cases, numerical methods are required.

### Lecture 20 MATHEMATICA® Example 3 Changing Variables in Symbolic Differential Equations

Download [notebooks](#), [pdf\(color\)](#), [pdf\(bw\)](#), or [html](#) from <http://pruffle.mit.edu/3.016-2007>.

The Bernoulli equation, Eq. 20-21, is used to demonstrate how to change variables in an ODE.

```

BernoulliEquation =
y'[x] + p[x] y[x] ==
r[x] (y[x]) ^ (a) 1

yRep = u[x] ^ (-1) 2
DyRep = D[yRep, x]

step1 =
BernoulliEquation /.
{y[x] -> yRep,
y'[x] -> DyRep} 3

step2 =
PowerExpand[step1] 4

step3 = Simplify[step2] 5

BE =
Solve[step3, u'[x]] 6

uprime = u'[x] /. BE 7

usol = u[x] /.
DSolve[u'[x] ==
uprime[[1]],
u[x], x] 8

ysol = (usol[[1]]) ^
(1 / (1 - a)) 9

BernoulliEquation
Simplify[p[x] ysol +
D[ysol, x]] 10

```

- 1: The Bernoulli equation is a non-linear first order ODE, but a series of transformations can turn it into an equivalent linear form.
- 2: Symbols for what will be used as replacements for  $y(x)$  and its derivative in *BernoulliEquation* are defined.
- 3: For `step1`, the symbols are used for a rule-replacement.
- 4: Using the form with replacements, the assumption that all variables are real is employed by using `PowerExpand`.
- 5: `Simplify` produces an equation for which the right-hand-side is zero; thus assuming that  $u(x)$  is not identically zero, it can be factored out of the equation.
- 6: Using `Solve` (n.b, not `DSolve`) to find  $u'(x)$  reveals the linear form of Bernoulli's equation in terms of the new variable.
- 7: The rule that is produced by `Solve` is used to extract the symbolic form of  $u'(x)$ ; the symbolic form of  $u'(x)$  is assigned to `uprime`.
- 8: To extract the solution (`usol`), we use the rule produced by `DSolve` on the equation  $u'(x) = \text{usol}$ .
- 9: The back-transformation is used to find the general solution  $y(x)$  to the non-linear form of the Bernoulli equation (`ysol`).
- 10: The solution, `ysol`, is plugged back into the left-hand-side of the Bernoulli equation and, with `Simplify`, is shown to be  $r(x)ysol^a$ .

## Lecture 20 MATHEMATICA® Example 4

## Numerical Solutions to Non-linear First-Order ODEs

Download [notebooks](#), [pdf\(color\)](#), [pdf\(bw\)](#), or [html](http://pruffle.mit.edu/3.016-2007) from <http://pruffle.mit.edu/3.016-2007>.

An example of computing the numerical approximation to the solution to a non-linear ODE is presented. The solutions are returned in the forms of a list of replacement rules to `InterpolatingFunction`. An `InterpolatingFunction` is a method to use numerical interpolation to extract an approximation for any point—it works just like a function and can be called on a variable like `InterpolatingFunction[0.2]`. In addition to the interpolation table, the definition specifies the domain over which the interpolation is considered valid.

Mathematica cannot find a direct solution to the following nonlinear ODE

```
DSolve[
Sin[2 Pi y'[x]^2] ==
y[x] x, y[x], x] 1
```

*NDSolve is a numerical method for finding a solution. An initial condition and the desired range of solution are required.*

```
solution =
NDSolve[{Sin[
2 Pi y'[x]^2] ==
y[x] x, y[0] == 1,
y, {x, 0, 3.5}}] 2
```

```
{y ->
InterpolatingFunction::
on[{{0., 3.5}},
<>], {y ->
InterpolatingFunction::
on[
{{0., 3.5}}, <>]}
```

```
y[0.5] /. solution 3
```

```
{0.907437,
1.09733 + 0. i}
```

```
y[Pi] /. solution 4
```

```
{0.0524983,
2.50186 - 0.61067 i}
```

- 1: This shows that `DSolve` cannot find a symbolic solution to  $\sin[2\pi(y')^2] = y(x)x$ .
- 2: Using `NDSolve` on a non-linear ODE, the solution is returned as a `InterpolatingFunction` replacement list. Note that there is a warning about “inverse functions” being used to find the solution; this is because of the `sin`-function which is causing Mathematica to assume a particular domain. There may be more solutions than the two that were that were returned as an `InterpolatingFunction`.
- 3–4: This demonstrates how the numerical approximation to the non-linear ODE is obtained at particular values of  $x$ .

## Lecture 20 MATHEMATICA® Example 5

## Plotting Numerical Solutions to Non-linear First-Order ODEs

Download [notebooks](#), [pdf\(color\)](#), [pdf\(bw\)](#), or [html](#) from <http://pruffle.mit.edu/3.016-2007>.

This is an example of how to extract plot-table expressions from the rules for `InterpolationFunctions` that are returned from `NDSolve`.

```

PStyle = {{Red, Thick},
{Darker[Green],
Thick}}; 1
PlotVanilla =
Plot[Evaluate[
y[x] /. solution],
{x, 0, 3.5},
PlotStyle -> PStyle,
PlotRange -> {0, 2},
PlotLabel -> "Plot"]; 2
PlotReal = Plot[
Evaluate[Re[y[x] /.
solution]],
{x, 0, 3.5}, 3
PlotStyle -> PStyle,
PlotLabel ->
"Real Part"];
PlotIm = Plot[
Evaluate[Im[y[x] /.
solution]],
{x, 0, 3.5}, 4
PlotStyle -> PStyle,
PlotLabel ->
"Imaginary Part"];
GraphicsRow[
{PlotVanilla,
PlotReal, PlotIm}, 5
ImageSize -> Small]

```

**1:** Because `solution` obtained above is a list containing two rules, two curves will be plotted. Here we define a short-hand for the expression that will be passed to `PlotStyle` in the plots below. The first curve will be red, and the second will be `Darker` green.

**2:** Here, `Plot` is called on the `y[x]` with replacements defined the rule-set for `InterpolatingFunctions`, `solution`, that was obtained from `NDSolve` previously. Using `Evaluate` here immediately creates a list of length two, and plot recognizes this as two curves to which the `PlotStyles` can be applied. If `Evaluate` were not used, then both curves would be red.

`Plot` only produces curves where the numerical value can be represented by a real number; if a solution has a point where it transforms from real to complex, `Plot` will show a curve that appears to end.

**3–4:** To determine the solution behavior, the real and imaginary parts are extracted with `Re` and `Im`.

**5:** This `GraphicsRow` indicates the solution behavior: the first solution is real over the domain where the interpolation is valid; the second solution transforms from real to complex near  $x = 0.8$ .