Lecture 22: Differential Operators, Harmonic Oscillators

Reading:
Kreyszig Sections: 2.3, 2.4, 2.7, 2.8, 2.9

Differential Operators

The idea of a function as "something" that takes a value (real, complex, vector, etc.) as "input" and returns "something else" as "output" should be very familiar and useful.

This idea can be generalized to *operators* that take a function as an argument and return another function.

The derivative operator operates on a function and returns another function that describes how the function changes:

$$\begin{aligned}
\mathcal{D}[f(x)] &= \frac{df}{dx} \\
\mathcal{D}[\mathcal{D}[f(x)]] &= \mathcal{D}^2[f(x)] = \frac{d^2 f}{dx^2} \\
\mathcal{D}^n[f(x)] &= \frac{d^n f}{dx^n} \\
\mathcal{D}[\alpha f(x)] &= \alpha D[f(x)] \\
\mathcal{D}[f(x) + g(x)] &= D[f(x)] + D[g(x)]
\end{aligned} \tag{22-1}$$

The last two equations above indicate that the "differential operator" is a linear operator.

The integration operator is the right-inverse of $\mathcal{D}$

$$\mathcal{D}[\mathcal{I}[f(x)]] = \mathcal{D}[\int f(x)dx] \tag{22-2}$$

but is only the left-inverse up to an arbitrary constant.

Consider the differential operator that returns a constant multiplied by itself

$$\mathcal{D}f(x) = \lambda f(x) \tag{22-3}$$

which is another way to write the the homogeneous linear first-order ODE and has the same form as an eigenvalue equation. In fact, $f(x) = \exp(\lambda x)$, can be considered an *eigenfunction* of Eq. 22-3.

For the homogeneous second-order equation,

$$\left(\mathcal{D}^2 + \beta\mathcal{D} - \gamma\right)[f(x)] = 0 \tag{22-4}$$

It was determined that there were two eigensolutions that can be used to span the entire solution space:

$$f(x) = C_+ e^{\lambda_+ x} + C_- e^{\lambda_- x} \tag{22-5}$$

Operators can be used algebraically, consider the inhomogeneous second-order ODE

$$\left(a\mathcal{D}^2 + b\mathcal{D} + c\right)[y(x)] = x^3 \tag{22-6}$$

By treating the operator as an algebraic quantity, a solution can be found[12]

$$\begin{aligned} y(x) &= \left(\frac{1}{a\mathcal{D}^2 + b\mathcal{D} + c}\right)[x^3] \\ &= \left(\frac{1}{c} - \frac{b}{c^2}\mathcal{D} + \frac{b^2 - ac}{c^3}\mathcal{D}^2 - \frac{b(b^2 - 2ac)}{c^3}\mathcal{D}^3 + \mathcal{O}(\mathcal{D}^4)\right) x^3 \\ &= \frac{x^3}{c} - \frac{3bx^2}{c^2} + \frac{6(b^2 - ac)x}{c^3} - \frac{6b(b^2 - 2ac)}{c^4} \end{aligned} \tag{22-7}$$

[12]This method can be justified by plugging back into the original equation and verifying that the result is a solution.

3.016 Home

Full Screen

Close

Quit

which solves Eq. 22-6.

The Fourier transform is also a linear operator:

$$\begin{aligned}\mathcal{F}[f(x)] =& g(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{\imath kx} dx \\ \mathcal{F}^{-1}[g(k)] =& f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} g(k) e^{-\imath kx} dk\end{aligned} \tag{22-8}$$

Combining operators is another useful way to solve differential equations. Consider the Fourier transform, $\mathcal{F}$, operating on the differential operator, $\mathcal{D}$:

$$\mathcal{F}[\mathcal{D}[f]] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \frac{df(x)}{dx} e^{\imath kx} dx \tag{22-9}$$

Integrating by parts,

$$= \frac{1}{\sqrt{2\pi}} f(x) \mid_{x=-\infty}^{x=\infty} - \frac{\imath k}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{\imath kx} dx \tag{22-10}$$

If the Fourier transform of $f(x)$ exists, then *typically*[13] $\lim_{x\rightarrow\pm\infty} f(x) = 0$. In this case,

$$\mathcal{F}[\mathcal{D}[f]] = -\imath k \mathcal{F}[f(x)] \tag{22-11}$$

and by extrapolation:

$$\begin{aligned}\mathcal{F}[\mathcal{D}^2[f]] &= -k^2 \mathcal{F}[f(x)] \\ \mathcal{F}[\mathcal{D}^n[f]] &= (-1)^n \imath^n k^n \mathcal{F}[f(x)]\end{aligned} \tag{22-12}$$

Operational Solutions to ODEs

Consider the heterogeneous second-order linear ODE which represent a forced, damped, harmonic oscillator that will be discussed later in this lecture.

$$M \frac{d^2 y(t)}{dt^2} + V \frac{dy(t)}{dt} + K_s y(t) = \cos(\omega_o t) \tag{22-13}$$

[13] It is not necessary that $\lim_{x\rightarrow\pm\infty} f(x) = 0$ for the Fourier transform to exist but it is satisfied in most every case. The condition that the Fourier transform exists is that $\int_{-\infty}^{\infty} |f(x)| dx$ exists and is bounded.

Apply a Fourier transform (mapping from the time (t) domain to a frequency (ω) domain) to both sides of 22-13:

$$\begin{aligned}\mathcal{F}[M\frac{d^2y(t)}{dt^2}+V\frac{dy(t)}{dt}+K_sy(t)] = \mathcal{F}[\cos(\omega_o t)]\\ -M\omega^2\mathcal{F}[y] - \imath\omega V\mathcal{F}[y] + K_s\mathcal{F}[y] = \sqrt{\frac{\pi}{2}}\,[\delta(\omega-\omega_o)+\delta(\omega+\omega_o)]\end{aligned} \tag{22-14}$$

because the Dirac Delta functions result from taking the Fourier transform of $\cos(\omega_o t)$.

Equation 22-14 can be solved for the Fourier transform:

$$\mathcal{F}[y] = \sqrt{\frac{-\pi}{2}}\frac{[\delta(\omega-\omega_o)+\delta(\omega+\omega_o)]}{M\omega^2+\imath\omega V - K_s} \tag{22-15}$$

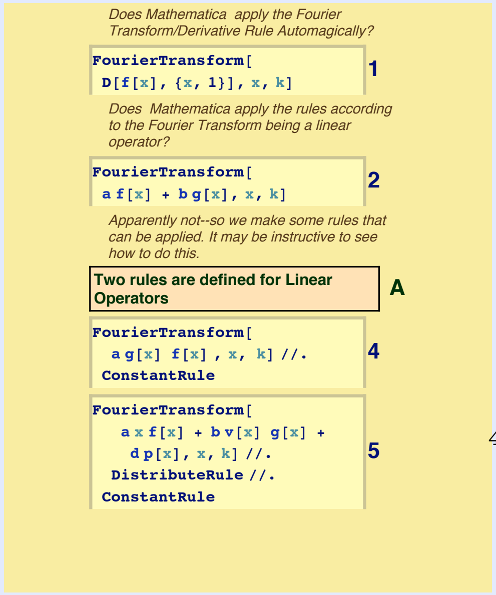
In other words, the particular solution Eq. 22-13 can be obtained by finding the function $y(t)$ that has a Fourier transform equal the the right-hand-side of Eq. 22-15–or, equivalently, operating with the inverse Fourier transform on the right-hand-side of Eq. 22-15.

MATHEMATICA® does have built-in functions to take Fourier (and other kinds of) integral transforms. However, using operational calculus to solve ODEs is a bit clumsy in MATHEMATICA® . Nevertheless, it may be instructive to force it—if only as an an example of using a good tool for the wrong purpose.

notebook (non-evaluated) pdf (evaluated, color) pdf (evaluated, b&w) html (evaluated)

Linear Operators and Derivatives

A check is made to see if `FourierTransform` obeys the rules of a linear operator (Eq. 22-1) and define rule-patterns for those cases where it doesn't.



Does Mathematica apply the Fourier Transform/Derivative Rule Automagically?

```
FourierTransform[
 D[f[x], {x, 1}], x, k]
```
1

Does Mathematica apply the rules according to the Fourier Transform being a linear operator?

```
FourierTransform[
 a f[x] + b g[x], x, k]
```
2

Apparently not--so we make some rules that can be applied. It may be instructive to see how to do this.

Two rules are defined for Linear Operators A

```
FourierTransform[
  a g[x] f[x] , x, k] //.
 ConstantRule
```
4

```
FourierTransform[
    a x f[x] + b v[x] g[x] +
     d p[x], x, k] //.
  DistributeRule //.
 ConstantRule
```
5

1: As of MATHEMATICA® 5.0, `FourierTransform` automatically implements Eqs. 22-12.

2: However, this will demonstrate that the "distribution-rule" isn't implemented automatically (n.b., although `Distribute` would implement this rule).

A: Define rules so that the FourierTransform acts as a linear functional operator (definitions suppressed in class-notes). *ConstantRule* is an example of a `RuleDelayed` (`:>`) that will allow replacement with patterns that will be evaluated when the rule is applied with `ReplaceAll` (`/.`); in this case, a `Condition` (`/;`) is appended to the rule so that those cofactors which don't depend on the transformation variable, x, can be identified with `FreeQ` and those that depend on x can be identified with `MemberQ`. *DistributeRule* uses `Distribute` to replace the Fourier transform of a sum with a sum of Fourier transforms.

4–5: The linear rules are dispatched by a `ReplaceRepeated` (`//.`) that will continue to use the replacement until the result stops changing. These are examples of $\mathcal{F}[ag(x)] = a\mathcal{F}[g(x)]$ and $\mathcal{F}[ag(x) + bh(x)] = a\mathcal{F}[g(x)] + b\mathcal{F}[h(x)]$.

3.016 Home

Full Screen

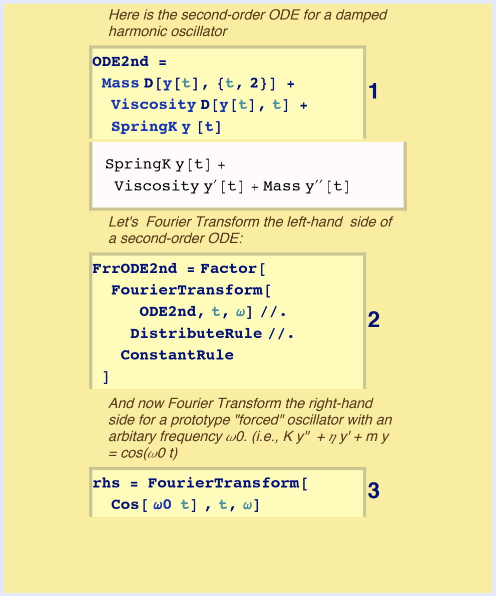
Close

Quit

notebook (non-evaluated) pdf (evaluated, color) pdf (evaluated, b&w) html (evaluated)

Fourier Transforming the Linear-Damped-Forced Harmonic Oscillator Equation into the Frequency Domain

The left- and right-hand sides of the damped harmonic oscillator ODE are Fourier transformed, producing an algebraic equation between the the solution in Fourier-space and the Fourier k-parameter.



Here is the second-order ODE for a damped harmonic oscillator

```
ODE2nd =
 Mass D[y[t], {t, 2}] +
  Viscosity D[y[t], t] +
  SpringK y [t]
```
1

```
SpringK y[t] +
 Viscosity y′[t] + Mass y″[t]
```

Let's Fourier Transform the left-hand side of a second-order ODE:

```
FrrODE2nd = Factor[
  FourierTransform[
     ODE2nd, t, ω] //.
    DistributeRule //.
   ConstantRule
 ]
```
2

And now Fourier Transform the right-hand side for a prototype "forced" oscillator with an arbitary frequency ω0. (i.e., K y" + η y' + m y = cos(ω0 t)

```
rhs = FourierTransform[
  Cos[ ω0 t] , t, ω]
```
3

1: This is the linear second-order for the "internal forcing" term of the harmonic oscillator. One could read this equation as Inertial Force+Frictional Force+Force to Restore to Minimal Potential Energy or $ma + \eta v + kx = m\ddot{x} + \eta\dot{x} + kx$

2: Fourier transforming (with `FourierTransform`) into the time domain converts the differential equation in the space domain into an algebraic equation in the time-domain.

3: If there is no "external force" on the harmonic oscillator, then the sum of the internal forces is zero. For the periodically-forced harmonic-oscillator, the right-hand-side of the equation can be expanded in a Fourier series. Here is a prototype of a right-hand-side, $\cos(\omega_o t)$, where ω_o is the forcing frequency. The forced-damped linear equation in the time-domain is obtained by transforming the external forces, or right-hand-side, of the harmonic oscillator.

3.016 Home

Full Screen

Close

Quit

notebook (non-evaluated) pdf (evaluated, color) pdf (evaluated, b&w) html (evaluated)

Fourier Transform Solution to the Damped-Forced Linear Harmonic Oscillator

The harmonic oscillator is algebraically solved in the time-domain, and then the solution is back-transformed into the real-space domain.

```
ftsol =
 Solve[FrrODE2nd == rhs,
  FourierTransform[
   y[t], t, ω]]
```
1

```
DampedHOAssumptions =
  ω0 > 0 && Mass > 0 &&
   Viscosity > 0 &&
   SpringK > 0;
```
2

```
FullSimplify[
 InverseFourierTransform[
  FourierTransform[y[t], t,
    ω] /. Flatten[ftsol],
  ω, t], Assumptions →
  DampedHOAssumptions]
```
3

```
GenSol = DSolve[
  Mass D[y[t], {t, 2}] +
    Viscosity D[y[t], t] +
    SpringK y[t] ==
   Cos[ω₀ t], y[t], t]
```
4

```
FullSimplify[
 y[t] /. Flatten[GenSol],
 Assumptions →
  DampedHOAssumptions]
```
5

1: `Solve` is used to find the algebraic solution to the Fourier-transformed solution to the harmonic oscillator.

2: *DampedHOAssumptions* is a collection of physical solution that will be passed to `FullSimplify`.

3: The real-space solution is obtained with `InverseFourierTransform` operating on the general form FourierTransform[y[t], t, ω] as a pattern-replacement for the rule obtained by `Solve`. This produces only the *particular solution* (i.e., the homogeneous solutions that depend on undetermined constants is not part of the particular solution.)

4–5: Here, `DSolve` is used to produced the full solution for comparison to the Fourier technique. It is the solution to the homogeneous equation plus the particular solution that was obtained by the Fourier transform method. The solution is extracted from the solution-rule and simplified with the *DampedHOAssumptions* .

3.016 Home

Full Screen

Close

Quit

Functionals and the Functions that Minimize Them: *Breaking the Cycle of Derivative and Function Minimization*

Equally powerful is the concept of a *functional* which takes a function as an argument and returns a value. For example $\mathcal{S}[y(x)]$, defined below, operates on a function $y(x)$ and returns its surface of revolution's area for $0 < x < L$:

$$\mathcal{S}[y(x)] = 2\pi \int_0^L y\sqrt{1+\left(\frac{dy}{dx}\right)^2}dx \tag{22-16}$$

This is the functional to be minimized for the question, "Of all surfaces of revolution that span from $y(x=0)$ to $y(x=L)$, which is the $y(x)$ that has the smallest surface area?"

This idea of finding "which function maximizes or minimizes something" can be very powerful and practical.

Suppose you are asked to run an "up-hill" race from some starting point $(x=0, y=0)$ to some ending point $(x=1, y=1)$ and there is a ridge $h(x,y) = x^2$. Of all the possible routes, which is the shortest route $y(x)$? The solution $y(x)$ is called the geodesic.

As an introductory example, we write a functional that returns a scalar length associated with a curve $y(x)$ that starts at x_b and terminates at x_b.

$$F[y(x)] = \int_{(x_b,y(x_b))}^{(x_e,y(x_e))} ds = \int_{(x_b,y(x_b))}^{(x_e,y(x_e))} \sqrt{dx^2+dy^2} = \int_{x_b}^{x_e}\sqrt{1+\left(\frac{dy}{dx}\right)}dx \tag{22-17}$$

In Equation 22-17, F takes any $y(x)$ (with some technical restrictions, such as integrability) and returns the arc-length associated with that $y(x)$ between two fixed points. The geodesic is defined by the function that minimizes Equation 22-17

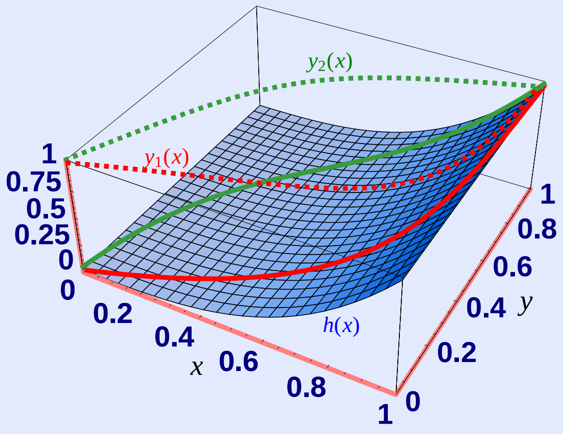


Figure 22-27: The terrain separating the starting point $(x = 0, y = 0)$ and ending point $(x = 1, y = 1)$. What is the shortest path between the starting and ending points? If the rate of climbing (or descending) is a known function of the slope, what is the quickest path? Assuming a model for how much running speed slows with the steepness of the path—which route would be quicker, one $(y_1(x))$ that starts going up-hill at first or another $(y_2(x))$ that initially traverses a lot of ground quickly?

These problems have some similarity to extrema in basic calculus—what is the parameter, variable, or point at which a given function is a maximum or a minimum. However, there is an important difference in the nature of the question that is being asked—what is the *function* that minimizes or maximizes a given functional. For basic calculus, the solution, or solutions, come from domain of possible solutions is the domain x over which the function $f(x)$ is defined (a line). In multivariable calculus, the solution(s) typically come from areas, volumes, and higher-dimensional spaces. For functionals, the solutions come from a "much larger" space. There is no obvious way to enumerate the set of trajectories that begin and end at a given point; such functions are uncountable.

The methods for finding such extremal functions derive from *variational calculus*, and the extension of basic calculus' derivative to functionals is called the *variational derivative.*

Introduction to Variational Calculus: Variation of Parameters

To introduce the idea of variational calculus, we will minimize the functional Equation 22-17, but only for an enumerable set of functions.

Suppose the starting point is $x_b, y(x_b)) = (0,0)$ and the ending point is $(1,1)$. Instead of choosing from *all functions* that connect the two points, we consider a smaller set of quadratic polynomials:

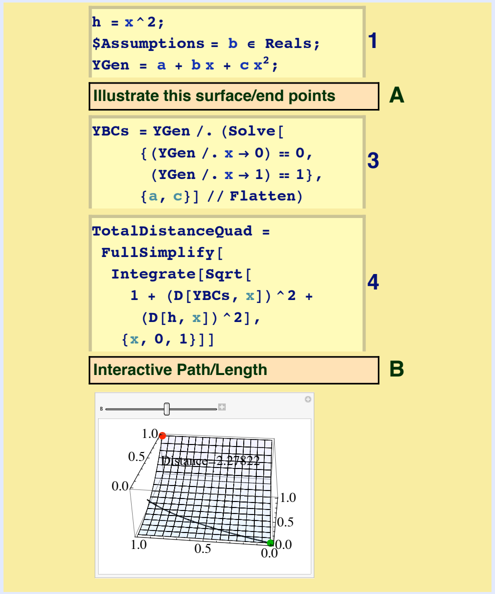
$$y(x) = a + bx + cx^2 \tag{22-18}$$

The two boundary conditions x_b and x_e constrain two of the three parameters (a, b, c). Inserting Equation 22-18 into Equation 22-17, the problem is reduced to a basic calculus problem of minimizing over a single variable (e.g., b if the boundary conditions are used to solve for a and c). This method is demonstrated in the following examples.

notebook (non-evaluated) pdf (evaluated, color) pdf (evaluated, b&w) html (evaluated)

Approximating the Geodesic

The quadratic polynomials (three parameters a, b, and c) is used to match boundary conditions, leaving a single parameter b. The constrained polynomial is used in the integral for total length, Equation 22-17.



```
h = x^2;
$Assumptions = b ∈ Reals;
YGen = a + b x + c x^2;
```
1

Illustrate this surface/end points A

```
YBCs = YGen /. (Solve[
     {(YGen /. x → 0) == 0,
      (YGen /. x → 1) == 1},
     {a, c}] // Flatten)
```
3

```
TotalDistanceQuad =
 FullSimplify[
  Integrate[Sqrt[
    1 + (D[YBCs, x])^2 +
     (D[h, x])^2],
   {x, 0, 1}]]
```
4

Interactive Path/Length B

1: The shape of the surface over which the trajectories is defined as a function of x, as is the general quadratic that will be used as the function for variation of parameters. Here, we use a kernel default-assumption, `$Assumptions`, that will be automatically for functions such as `Integrate` and `Simplify`.

A: A graphic, *TheSurface* , is constructed with indicated initial- and end-points.

3: Here, the quadratic approximation is constrained to its initial- and end-points with replacement and the rule produced by `Solve`.

4: The will produce a closed form for the total distance as a function of a single parameter, b

B: `Manipulate` is used to produce an interactive graphic that illustrates the quadratic approximation as a function of b and the computed length.

3.016 Home

Full Screen

Close

Quit

Variation of Parameters for the Geodesic Approximation

This example shows that the quadratic approximation obtained by variation of parameters is close to the exact geodesic that is calculated by the calculus of variations. The method to find the exact geodesic is described below.

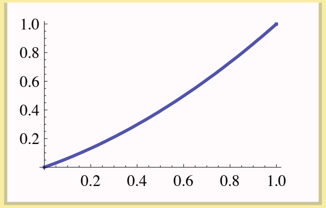
```
Plot[TotalDistanceQuad,
 {b, -2, 6}]
```
1

```
BminsolGeodesicQuad =
 FindMinimum[
  TotalDistanceQuad,
  {b, 0, 1}]
```
2

Use the minimizing b to find the approximation.

```
GeodesicQuadSolution =
 YBCs /.
  BminsolGeodesicQuad[[2]]
```
3

```
GeodesicQuadPlot =
 Plot[GeodesicQuadSolution,
  {x, 0, 1},
  PlotStyle → Thick]
```
4



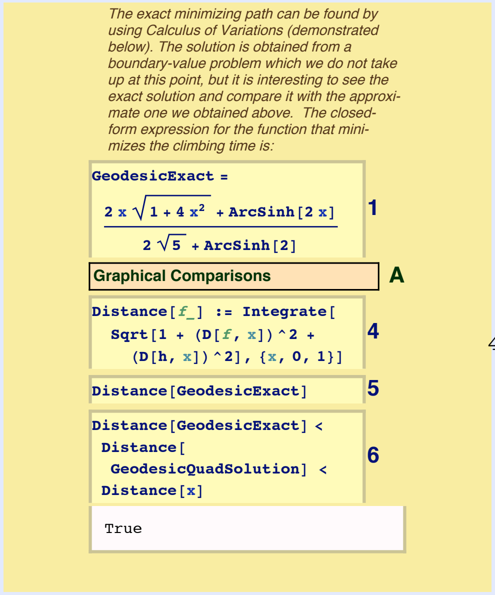
1: Plotting the remaining parameter shows that a minimum exists.

2: `FindMinimum` returns a list with the minimal value and a rule for the minimizing b.

3–4: Using the minimizing rule for b, we can replace b in the constrained quadratic approximation and plot it. This is the quadratic approximation to the geodesic for the given surface and boundary conditions.

Comparison of the Approximation to the Exact Geodisic

The quadratic polynomial is shown to have a minimum length with respect to a single unconstrained parameter. The minimizing approximation is computed and visualized.



The exact minimizing path can be found by using Calculus of Variations (demonstrated below). The solution is obtained from a boundary-value problem which we do not take up at this point, but it is interesting to see the exact solution and compare it with the approximate one we obtained above. The closed-form expression for the function that minimizes the climbing time is:

```
GeodesicExact =
```

$$\frac{2\,x\,\sqrt{1+4\,x^2}+\text{ArcSinh}[2\,x]}{2\,\sqrt{5}+\text{ArcSinh}[2]}$$ 1

Graphical Comparisons A

```
Distance[f_] := Integrate[
  Sqrt[1 + (D[f, x])^2 +
    (D[h, x])^2], {x, 0, 1}]
```
4

```
Distance[GeodesicExact]
```
5

```
Distance[GeodesicExact] <
 Distance[
  GeodesicQuadSolution] <
 Distance[x]
```
6

```
True
```

1: *GeodesicExact* is the exact geodesic for the specified surface ($h = x^2$) and boundary conditions. (This calculation is provided in a subsequent example).

A: Visual comparisons between the approximation and the exact solution show that the approximation is quite good. This is not a general rule, and we cannot know in advance if an approximation by variation of parameters will be good or not.

4–5: The functional is encoded in this `Distance` function, which takes a function of x as an argument and integrates over $0 < x < 1$.

6: This shows that the geodesic is shorter than the approximation, and the approximation is shorter than a straight line ($y(x) = x$) projected onto the x–y plane.

3.016 Home

Full Screen

Close

Quit

Shortest Time Paths: The Brachiostone

The geodesic gave the shortest-distance path between two points—a related question is, "Given a velocity, what is the quickest (shortest time) path between two points?" The answer is related the *brachistochrone* which is the path of most rapid descent with constant acceleration. I don't know what to call the shortest-time path, so I am making up a name " *brachiostone*". Perhaps a better name would be the *Fermatic*, becase the curve is related to a generalized Fermat's theorem. However, this could be confused with *fermata* which is a pause of unspecified length; so I suppose that *MiniFermatizoid* might be the best choice of all. However, in future editions to these notes, I am going to change the name to *Brakkes' Chrone*, in honor of one of my heros, *Ken Brakke* http://www.susqu.edu/brakke/. Neologisms are so entertaining—and a delightful waste of time.

For traversing a hill, it is a reasonable model for running speed to be a decreasing function of climbing-angle α, and to have the speed fall to zero when the trajectory is "straight-up." Thus, we select a model such as

$$v(s) = \cos(\alpha(s)) \tag{22-19}$$

where s is the arclength along the path. The maximum speed occurs on flat ground $\alpha = 0$ and running speed monotonically falls to zero as $\alpha \to \pi/2$. To calculate the time required to traverse *any* path $y(x)$ with endpoints $y(0) = 0$ and $y(1) = 1$,

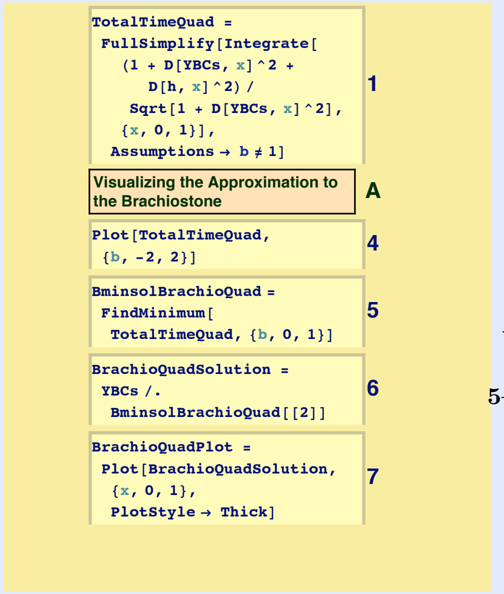
$$\begin{aligned}
\frac{ds}{dt} = v(s) = \cos(\alpha(s)) &= \frac{\text{local horizontal}}{\text{local arclength}} = \frac{\sqrt{dx^2 + dy^2}}{\sqrt{dx^2 + dy^2 + dh^2}} \\
\text{therefore } dt = \frac{ds}{v(s)} &= \frac{dx^2 + dy^2 + dz^2}{\sqrt{dy^2 + dx^2}} = \frac{1 + \frac{dy}{dx}^2 + \frac{dh}{dx}^2}{\sqrt{1 + \frac{dy}{dx}^2}}\, dx \\
\text{therefore time}[y(x)] &= \int_{x_b}^{x_e} \frac{1 + \frac{dy}{dx}^2 + \frac{dh}{dx}^2}{\sqrt{1 + \frac{dy}{dx}^2}}\, dx
\end{aligned} \tag{22-20}$$

The hill $h(x) = x^2$ can be inserted into Equation 22-20 for the time as a functional of the path between fixed points $(0, 0, 0)$ and $(1, 1, 1)$.

Lecture 22 MATHEMATICA® Example 7

Approximating the Brachiostone by Variation of Parameters

The same method for finding an approximation to the geodesic is applied to the minimum-time functional in Equation 22-20. (See preceding text on definition of brachiostone.)



```
TotalTimeQuad =
 FullSimplify[Integrate[
   (1 + D[YBCs, x]^2 +
      D[h, x]^2) /
    Sqrt[1 + D[YBCs, x]^2],
   {x, 0, 1}],
  Assumptions → b ≠ 1]                      1
```

Visualizing the Approximation to the Brachiostone A

```
Plot[TotalTimeQuad,
 {b, -2, 2}]                                4

BminsolBrachioQuad =
 FindMinimum[
  TotalTimeQuad, {b, 0, 1}]                 5

BrachioQuadSolution =
 YBCs /.
  BminsolBrachioQuad[[2]]                   6

BrachioQuadPlot =
 Plot[BrachioQuadSolution,
  {x, 0, 1},
  PlotStyle → Thick]                        7
```

1: The same quadratic (constrained to the boundary conditions) as was used for the geodesic is utilized for the brachiostone (Equation 22-20). In this case, there is a closed-form solution for the undetermined parameter, but this not typical for other functionals.

A: The brachiostone approximation is visualized by superposing onto the "hill" with the exact geodesic.

4: Plotting the time as a function of b indicates that there is a minimizing b.

5–7: The minimizing b is inserted back into the quadratic approximation

3.016 Home
Full Screen
Close
Quit

Introduction to Calculus of Variations

Suppose the functional depends on one function of single variable, $y(x)$, and its derivative, $y'(x)$. Furthermore, consider the fixed end-point problem (i.e., $y(x_b) = y_b$ and $y(x_e) = y_e$ are specified.

The general form of the functional is:

$$F[y(x)] = \int_{x_b}^{x_e} f[x, y(x), y'(x)]\, dx \tag{22-21}$$

We want to introduce a notation for functions that are "nearby" to a function $y(x)$[14] To do this, let a function near to $y(x)$ be described as $y(x) + v(x)\Delta t$. (It may be useful to think of t as a time-like variable and $v(x)$ is the instantaneous local velocity away from $y(x)$; but, generally, t, could be any scalar parameter.) Because $y(x)$ is assumed to match the boundary conditions, any admissible variation $y + v\Delta t$ must also match the boundary conditions, so the 'velocity' $v(x)$ at the boundaries must vanish. Therefore, for functions near to $y(x)$,

$$F[y + v\Delta t] = \int_{x_b}^{x_e} f[x, y(x) + v(x)\Delta t, y'(x) + v'(x)\Delta t]\, dx \tag{22-22}$$

Both sides depend on the scalar quantity Δt, and so we will expand about $\Delta t = 0$. We treat the integrand $f(x, y, y')$ as a function of three variables (it is after all, because f is being evaluated point-wise in the integral). Therefore, partial derivative must appear in the expansion:

$$F[y] + \frac{\delta F}{\delta y}\bigg|_{\Delta t=0} v\Delta t = \int_{x_b}^{x_e} f[x, y(x), y'(x)]\, dx + \int_{x_b}^{x_e} \left[\frac{\partial f}{\partial y} v(x) + \frac{\partial f}{\partial y'} v'(x)\right]\bigg|_{\Delta t=0} \Delta t\, dx \tag{22-23}$$

where we use a "δ" to indicate the *variational derivative* of a functional. Canceling common terms and integrating by parts,

$$\frac{\delta F}{\delta y}\bigg|_{\Delta t=0} v\Delta t = \Delta t \left\{ y(x)v(x)\bigg|_{x_b}^{x_e} + \int_{x_b}^{x_e} \left[\left(\frac{\partial f}{\partial y} - \frac{d}{dx}\frac{\partial f}{\partial y'}\right)\bigg|_{\Delta t=0} v(x)\right] dx \right\} \tag{22-24}$$

[14] A precise definition of "closeness" of functions is somewhat arbitrary and depends on the 'norm' defined for functions (and because gradients have a length and a direction, variational gradients also depend on the norm). Typically, variational calculus is introduced with the *l2-norm*, $f(x) \cdot g(x) \equiv \int f(x)g(x)dx$, and the resulting variation becomes $\delta F \cdot v\Delta t = \int [\partial F/\partial y - (d/dx)\partial F/\partial y']v\Delta t$ which, for any h that satisfies the boundary conditions, can equal zero only if the integrand of the variation vanishes (i.e, if the variation is 'orthogonal' to an arbitrary h. For several applications of other norms to materials science, see "Variational methods for microstructural-evolution theories", Carter W.C., Taylor J.E, Cahn J.W., JOM (Journal of the Materials Soc.), 49(12) 30–36, 1997

Because $v(x)$ must vanish at the end-points, and because the terms that are being evaluated at $t = 0$ do not depend on t, then

$$\frac{\delta F}{\delta y} \cdot v = \int_{x_b}^{x_e} \left[\frac{\partial f}{\partial y} - \frac{d}{dx}\frac{\partial f}{\partial y'} \right] v(x)\, dx \tag{22-25}$$

Because $v(x)$ is arbitrary (except for satisfying the boundary conditions), the only way that the functional derivative can vanish is for

$$\frac{\partial f}{\partial y} - \frac{d}{dx}\frac{\partial f}{\partial y'} = 0 \tag{22-26}$$

which is called the *Euler equation* and is the condition for a functional to be extremal with respect to a variation of its function-argument, $y(x)$.

We could also think of Equation 22-25 as representing the integral-sum of the instantaneous changes in the scalar value of the functional as its function $y(x)$ changes. The functional is stationary (i.e., a necessary condition for an extremum) if the variational derivative vanishes everywhere on $x_b < x < x_e$. Because we have a condition as a function of a single variable, the form of Euler's equation in Equation 22-26 is an ordinary differential equation of derivatives of $y(x)$ in x.

For example, consider the geodesic problem from the above example on the surface $h(x) = x^2$, with fixed end-points $y(x = 0) = 0$ and $y(x = 1) = 1$. The functional is

$$F[y(x)] = \int_0^1 \sqrt{1 + \frac{dy}{dx}^2 + \frac{dh}{dx}^2}\, dx = \int_0^1 \sqrt{1 + \frac{dy}{dx}^2 + 4x^2}\, dx \tag{22-27}$$

therefore,

$$\begin{aligned} \frac{\partial f}{\partial y} =& 0 \text{ and } \frac{\partial f}{\partial y'} = \frac{\frac{dy}{dx}}{\sqrt{1 + 4x^2 + \frac{dy}{dx}^2}} \\ \frac{d}{dx}\frac{\partial f}{\partial y'} =& \frac{\frac{d^2y}{dx^2}}{\sqrt{1 + 4x^2 + \frac{dy}{dx}^2}} - \frac{\frac{dy}{dx}\left(8x + 2\frac{dy}{dx}\frac{d^2y}{dx^2}\right)}{\left(1 + 4x^2 + \frac{dy}{dx}^2\right)^{3/2}} \\ =& \frac{\left(1 + 4x^2\right)\frac{d^2y}{dx^2} - 4x\frac{dy}{dx}}{\left(1 + 4x^2 + \frac{dy}{dx}^2\right)^{3/2}} \end{aligned} \tag{22-28}$$

3.016 Home

Full Screen

Close

Quit

©W. Craig Carter

The Euler equation becomes

$$\frac{4x\frac{dy}{dx} - (1+4x^2)\frac{d^2y}{dx^2}}{(1+4x^2+\frac{dy}{dx}^2)^{3/2}} = 0 \qquad \text{(22-29)}$$

The numerator can be set equal to zero, and the result is an integrable second-order linear ODE.

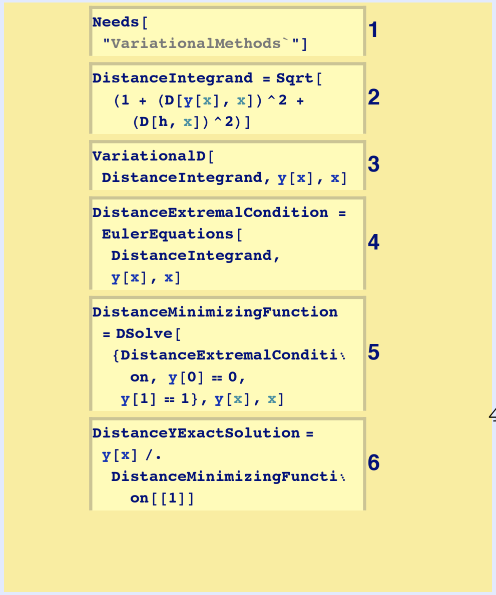
This and the example for the brachiostone is demonstrated in the following examples.

Lecture 22 MATHEMATICA® Example 8

notebook (non-evaluated) pdf (evaluated, color) pdf (evaluated, b&w) html (evaluated)

Euler's equation and Exact Solution to Geodesic

The variational derivative from the package `VariationalMethods` is used with `DSolve` to calculate the exact geodesic for which an approximate solution was found above.



```
Needs[
 "VariationalMethods`"]                         1

DistanceIntegrand = Sqrt[
   (1 + (D[y[x], x])^2 +
     (D[h, x])^2)]                              2

VariationalD[
 DistanceIntegrand, y[x], x]                    3

DistanceExtremalCondition =
 EulerEquations[
  DistanceIntegrand,
  y[x], x]                                      4

DistanceMinimizingFunction
 = DSolve[
  {DistanceExtremalConditi
    on, y[0] == 0,
   y[1] == 1}, y[x], x]                         5

DistanceYExactSolution =
 y[x] /.
  DistanceMinimizingFuncti
    on[[1]]                                     6
```

1: The `VariationalMethods` package has functions for many methods in the calculus of variations. We will use only a few of the simpler methods in this and the following example.

2: *DistanceIntegrand* is the integrand for total length; it will play the role of $f[x, y, y']$ in Equation 22-26

3: `VariationalD` computes the right-hand-side of the expression in Equation 22-25.

4: `EulerEquations` gives the equations, for a given integrand, for the extremal solution. *DistanceExtremalCondition* will be the ordinary differential equation, Equation 22-28.

4–5: Using `DSolve` to solve Euler's equation with the boundary conditions produces the exact solution.

3.016 Home

Full Screen

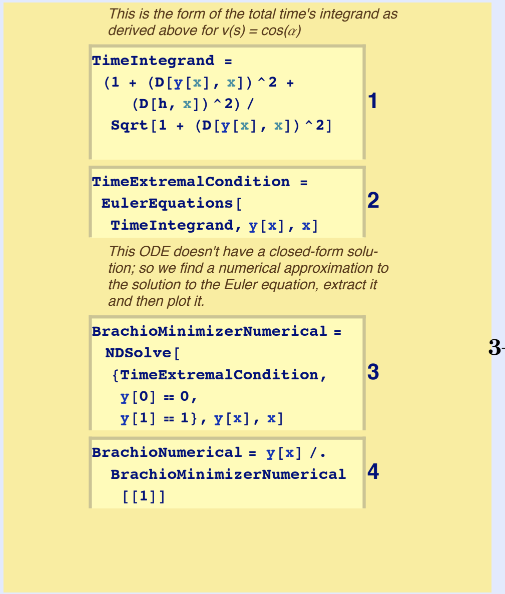
Close

Quit

notebook (non-evaluated) pdf (evaluated, color) pdf (evaluated, b&w) html (evaluated)

Euler's equation and Numerical Solution to Brachiostone

The Euler's equation for the total-time function defined in Equation 22-20 is solved by numerical methods.



This is the form of the total time's integrand as derived above for v(s) = cos(α)

```
TimeIntegrand =
 (1 + (D[y[x], x])^2 +
    (D[h, x])^2) /
  Sqrt[1 + (D[y[x], x])^2]
```
1

```
TimeExtremalCondition =
 EulerEquations[
  TimeIntegrand, y[x], x]
```
2

This ODE doesn't have a closed-form solution; so we find a numerical approximation to the solution to the Euler equation, extract it and then plot it.

```
BrachioMinimizerNumerical =
 NDSolve[
  {TimeExtremalCondition,
   y[0] == 0,
   y[1] == 1}, y[x], x]
```
3

```
BrachioNumerical = y[x] /.
  BrachioMinimizerNumerical
   [[1]]
```
4

1: *TimeIntegrand* is the term that was derived in Equation 22-20 and used in the approximate method for the brachiostone calculation.

2: The Euler equations for this integrand produces a non-linear second-order ODE that doesn't have a closed form solution.

3–4: However, a numerical solver `NDSolve` can be employed. In this case `NDSolve` runs into a few numerical difficulties around $x = 0.5$, but it produces a very reasonable solution that 'beats' the approximation given above.

3.016 Home

Full Screen

Close

Quit

Visualizing the Brachiostone and Comparison to the Approximation Obtained by Variation of Parameters

The numerical solution obtained above is plotted and compared to the approximate solution.

```
BrachioExactPlot =
 Plot[BrachioNumerical,
  {x, 0, 1}, PlotStyle →
   {Thick, Darker[Green]}]                    1

GraphicsRow[
 {Show[BrachioQuadPlot,
   BrachioExactPlot] ,
  Show[ Plot[
    BrachioNumerical -
     BrachioQuadSolution,
    {x, 0, 1}, PlotStyle →
     {Thickness[0.005],
      Hue[1]}]]}]                             2

Time[f_] :=
 Integrate[TimeIntegrand /.
   y'[x] → D[f, x],
  {x, 0, 1.0}]                                3

N[Time[BrachioNumerical]] <
 Chop[Time[
   BrachioQuadSolution]] <
 Time[x]                                      4

True
```

1–2: This visualized the computed brachiostone. It indicates that a better (and reasonable) strategy that it is advantageous to run up-hill when the slope is small, and then traverse over a longer distance with reduced slope (as in a "switch-back" in a hiking trail). In this case, the quadratic approximation is still quite good, but not as good as in the geodesic.

3: This function takes a function of x for an argument and returns the total time assuming the $v(s) = \cos\alpha(s)$ model on a hill given by $h = x^2$.

4: This demonstrate that the numerical solution to the Euler equation has a shorter time than the quadratic approximation which, in turn, is shorter than the "projected-straight-line." In this inequality, we use `N` with `Integrate` which is equivalent to using `NIntegrate`. The quadratic solution has a small imaginary part that arises from numerical imprecision—this is removed with `Chop`.

3.016 Home

Full Screen

Close

Quit

Harmonic Oscillators

There is a set of three well-worked examples, worked out in increasing levels of sophistication located at http://pruffle.mit.edu/3.016/mathematica-paradigms.html. These could be useful as a future resource.

Methods for finding general solution to the linear inhomogeneous second-order ODE

$$a\frac{d^2y(t)}{dt^2} + b\frac{dy(t)}{dt} + cy(t) = F(t) \tag{22-30}$$

have been developed and worked out in MATHEMATICA® examples.

Eq. 22-30 arises frequently in physical models, among the most common are:

$$\begin{aligned} \text{Electrical circuits:} \quad & L\frac{d^2I(t)}{dt^2} + \rho l_o\frac{dI(t)}{dt} + \frac{1}{C}I(t) = V(t) \\ \text{Mechanical oscillators:} \quad & M\frac{d^2y(t)}{dt^2} + \eta l_o\frac{dy(t)}{dt} + K_s y(t) = F_{app}(t) \end{aligned} \tag{22-31}$$

where:

	Mechanical	Electrical
Second Order	**Mass M**: Physical measure of the ratio of momentum field to velocity	**Inductance L**: Physical measure of the ratio of stored magnetic field to current
First Order	**Drag Coefficient $c = \eta l_o$** (η is viscosity l_o is a unit displacement): Physical measure of the ratio environmental resisting forces to velocity—or proportionality constant for energy dissipation with square of velocity	**Resistance $R = \rho l_o$** (ρ is resistance per unit material length l_o is a unit length): Physical measure of the ratio of voltage drop to current—or proportionality constant for power dissipated with square of the current.
Zeroth Order	**Spring Constant K_s**: Physical measure of the ratio environmental force developed to displacement—or proportionality constant for energy stored with square of displacement	**Inverse Capacitance $1/C$**: Physical measure of the ratio of voltage storage rate to current—or proportionality constant for energy storage rate dissipated with square of the current.
Forcing Term	**Applied Voltage $V(t)$**: Voltage applied to circuit as a function of time.	**Applied Force $F(t)$**: Force applied to oscillator as a function of time.

For the homogeneous equations (i.e. no applied forces or voltages) the solutions for physically allowable values of the coefficients can either be oscillatory, oscillatory with damped amplitudes, or, completely damped with no oscillations. (See Figure 21-25). The homogeneous equations are sometimes called *autonomous* equations—or *autonomous systems*.

Simple Undamped Harmonic Oscillator

The simplest version of a homogeneous Eq. 22-30 with no damping coefficient ($b = 0$, $R = 0$, or $\eta = 0$) appears in a remarkably wide variety of physical models. This simplest physical model is a simple harmonic oscillator—composed of a mass accelerating with a linear spring restoring force:

$$\begin{aligned}\text{Inertial Force} &= \text{Restoring Force}\\ M\text{Acceleration} &= \text{Spring Force}\\ M\frac{d^2y(t)}{dt^2} &= -K_s y(t)\\ M\frac{d^2y(t)}{dt^2} + K_s y(t) &= 0\end{aligned} \tag{22-32}$$

Here y is the displacement from the equilibrium position–i.e., the position where the force, $F = -dU/dx = 0$. Eq. 22-32 has solutions that oscillate in time with frequency ω:

$$\begin{aligned} y(t) &= A\cos\omega t + B\sin\omega t\\ y(t) &= C\sin(\omega t + \phi)\end{aligned} \tag{22-33}$$

where $\omega = \sqrt{K_s/M}$ is the natural frequency of oscillation, A and B are integration constants written as amplitudes; or, C and ϕ are integration constants written as an amplitude and a phase shift.

The simple harmonic oscillator has an *invariant*, for the case of mass-spring system the invariant is the total energy:

$$\begin{aligned}\text{Kinetic Energy} + \text{Potential Energy} &=\\ \frac{M}{2}v^2 + \frac{K_s}{2}y^2 &=\\ \frac{M}{2}\frac{dy}{dt}^2 + \frac{K_s}{2}y^2 &=\\ A^2\omega^2\frac{M}{2}\cos^2(\omega t+\phi) + A^2\frac{K_s}{2}\sin^2(\omega t+\phi) &=\\ A^2(\omega^2\frac{M}{2}\cos^2(\omega t+\phi) + \frac{M\omega^2}{2}\sin^2(\omega t+\phi) &=\\ A^2M\omega^2 &= \text{constant}\end{aligned} \tag{22-34}$$

There are a remarkable number of physical systems that can be reduced to a simple harmonic oscillator (i.e., the model can be reduced to Eq. 22-32). Each such system has an analog to a mass, to a spring constant, and thus to a natural frequency.

Furthermore, every such system will have an invariant that is an analog to the total energy—an in many cases the invariant will, in fact, be the total energy.

The advantage of reducing a physical model to a harmonic oscillator is that *all of the physics follows from the simple harmonic oscillator.*

Here are a few examples of systems that can be reduced to simple harmonic oscillators:

Pendulum By equating the rate of change of angular momentum equal to the torque, the equation for pendulum motion can be derived:

$$MR^2\frac{d^2\theta}{dt^2} + MgR\sin\theta = 0 \tag{22-35}$$

for small-amplitude pendulum oscillations, $\sin(\theta) \approx \theta$, the equation is the same as a simple harmonic oscillator.

It is instructive to consider the invariant for the non-linear equation. Because

$$\frac{d^2\theta}{dt^2} = \frac{d\theta}{dt}\left(\frac{d\frac{d\theta}{dt}}{d\theta}\right) \tag{22-36}$$

Eq. 22-35 can be written as:

$$MR^2\frac{d\theta}{dt}\left(\frac{d\frac{d\theta}{dt}}{d\theta}\right) + MgR\sin(\theta) = 0 \tag{22-37}$$

$$\frac{d}{d\theta}\left[\frac{MR^2}{2}\left(\frac{d\theta}{dt}\right)^2 - MgR\cos(\theta)\right] = 0 \tag{22-38}$$

which can be integrated with respect to θ:

$$\frac{MR^2}{2}\left(\frac{d\theta}{dt}\right)^2 - MgR\cos(\theta) = \text{constant} \tag{22-39}$$

This equation will be used as a level-set equation to visualize pendulum motion.

Buoyant Object Consider a buoyant object that is slightly displaced from its equilibrium floating position. The force (downwards) due to gravity of the buoy is $\rho_{bouy} g V_{bouy}$ The force (upwards) according to Archimedes is $\rho_{water} g V_{sub}$ where V_{sub} is the volume of the buoy that is submerged. The equilibrium position must satisfy $V_{sub-eq}/V_{bouy} = \rho_{bouy}/\rho_{water}$.

If the buoy is slightly perturbed at equilibrium by an amount δx the force is:

$$\begin{aligned} F =& \rho_{water} g (V_{sub-eq} + \delta x A_o) - \rho_{buoy} g V_{buoy} \\ F =& \rho_{water} g \delta x A_o \end{aligned} \tag{22-40}$$

where A_o is the cross-sectional area at the equilibrium position. Newton's equation of motion for the buoy is:

$$M_{buoy} \frac{d^2 y}{dt^2} - \rho_{water} g A_o y = 0 \tag{22-41}$$

so the characteristic frequency of the buoy is $\omega = \sqrt{\rho_{water} g A_o / M_{bouy}}$.

Single Electron Wave-function The one-dimensional Schrödinger equation is:

$$\frac{d^2 \psi}{dx^2} + \frac{2m}{\hbar^2} \left(E - U(x) \right) \psi = 0 \tag{22-42}$$

where $U(x)$ is the potential energy at a position x. If $U(x)$ is constant as in a free electron in a box, then the one-dimensional wave equation reduces to a simple harmonic oscillator.

In summation, just about any system that oscillates about an equilibrium state can be reduced to a harmonic oscillator.

Index

`/.`, 286
`//.`, 286
`/;`, 286
`:>`, 286
`$Assumptions`, 292

autonomous systems, 304

brachiostone, 295
brachistochrone, 295
Brakkes' Chrone, 295

`Chop`, 302
`Condition`, 286
ConstantRule, 286

DampedHOAssumptions, 288
damping factors, 304
differential operators, 282
Dirac delta function
  Fourier transforms, 285
DistanceExtremalCondition, 300
DistanceIntegrand, 300
`Distribute`, 286
DistributeRule, 286
`DSolve`, 288, 300

eigenfunction, 283
eigenvalue
  of an operator, 283
electrical circuits as harmonic oscillators, 303
Euler equation, 298
`EulerEquations`, 300
Example function
  ConstantRule, 286
  DampedHOAssumptions, 288
  DistanceExtremalCondition, 300
  DistanceIntegrand, 300
  DistributeRule, 286
  GeodesicExact, 294
  TheSurface, 292
  TimeIntegrand, 301
extremal functions, 289

fermata, 295
Fermatic, 295
`FindMinimum`, 293
forces
  in harmonic oscillator model, 305
Fourier transform
  as a linear operator, 284
  as a method to solve ODEs, 284
`FourierTransform`, 286, 287
`FreeQ`, 286
frequency
  harmonic oscillator, 305
`FullSimplify`, 288
functionals, 289
functions of functions, 289

3.016 Home

Full Screen

Close

Quit

geodesic, 289
GeodesicExact, 294

harmonic oscillator
  buoys, 306
  damped forced ODE, 284
  force interpretation, 287
  Fourier transform of, 287
  pendulum, 306
  single electron wave function, 307
  solutions using Fourier transforms, 284
harmonic oscillators, 303
  instances in different physical phenomena, 303

`Integrate`, 292
invariant
  in harmonic oscillator, 305
inverse Fourier transform, 285
`InverseFourierTransform`, 288

Ken Brakke, 295

l2-norm, 297
length minimizing path
  geodesic, 289
linear operators
  defining rules for, 286
linear transformations
  defining rules for, 286

`Manipulate`, 292
Mathematica function
  /., 286
  //., 286
  /;, 286
  :¿, 286
  Chop, 302
  Condition, 286
  DSolve, 288, 300
  Distribute, 286
  EulerEquations, 300
  FindMinimum, 293
  FourierTransform, 286, 287
  FreeQ, 286
  FullSimplify, 288
  Integrate, 292
  InverseFourierTransform, 288
  Manipulate, 292
  MemberQ, 286
  NDSolve, 301
  ReplaceAll, 286
  ReplaceRepeated, 286
  RuleDelayed, 286
  Simplify, 292
  Solve, 288, 292
  VariationalD, 300
  $Assumptions, 292
Mathematica package
  VariationalMethods, 300
`MemberQ`, 286
MiniFermatizoid, 295

`NDSolve`, 301
NIntegrate
  equivalence to N and Integrate, 302

operators
 algebraic operations on, 283
 as methods to solve ODEs, 284
 differential, 282
optimal path, 293

particular solution, 288

`ReplaceAll`, 286
`ReplaceRepeated`, 286
RLC circuits, 303
`RuleDelayed`, 286

`Simplify`, 292
`Solve`, 288, 292
surface of revolution, 289
switch-backs
 hiking, 302

TheSurface, 292
time domain and frequency domain, 285
TimeIntegrand, 301

variational calculus, 290
variational derivative, 291, 297
`VariationalD`, 300
`VariationalMethods`, 300