Lecture 16: Integral Theorems

Reading:
Kreyszig Sections: 10.8, 10.9 (pages463–467, 468–473)

Higher-dimensional Integrals

The fundamental theorem of calculus was generalized in a previous lecture from an integral over a single variable to an integration over a region in the plane. Specifically, for generalizing to Green's theorem in the plane, a vector derivative of a function integrated over a line and evaluated at its endpoints was generalized to a vector derivative of a function integrated over the plane.
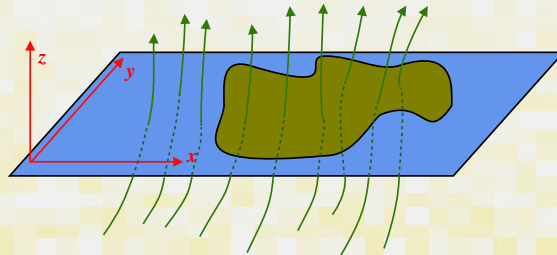

Figure 16-15: Illustrating how Green's theorem in the plane works. If a *known* vector function is integrated over a region in the plane then that integral should only depend on the bounding curve of that region.
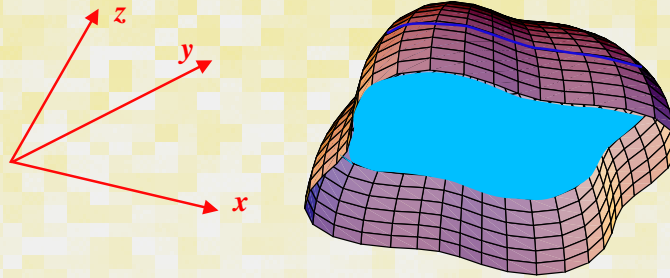
Figure 16-16: Illustration of a generalization to the Green's theorem in the plane: Suppose there is a bowl of a known shape submerged in a fluid with a trapped bubble. The bubble is bounded by two different surfaces, the bowl down to $z = 0$ and the planar liquid surface at that height. Integrating the function $\int_{V_B} dV$ over the bubble gives its volume. The volume must also be equal to an integral $\int \int_{\partial V_B} z \, dx dy$ over the (oriented) surface of the liquid. However, the volume of bubble can be determined from only the curve defined by the intersection of the bowl and the planar liquid surface; so the volume must also be equal to $\oint_C (\text{some function}) ds$.

The Divergence Theorem

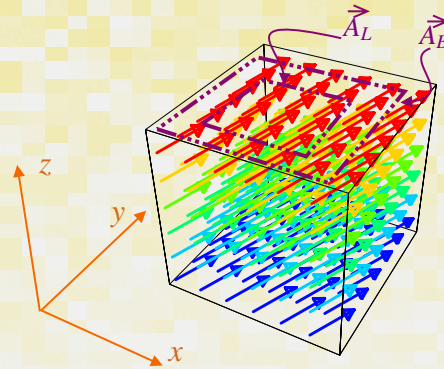Suppose there is "stuff" flowing from place to place in three dimensions.

Figure 16-17: Illustration of a vector "flow field" $\vec{J}$ near a point in three dimensional space. If each vector represents the rate of "stuff" flowing per unit area of a plane that is normal to the direction of flow, then the dot product of the flow field integrated over a planar oriented area $\vec{A}$ is the rate of "stuff" flowing through that plane. For example, consider the two areas indicated with purple (or dashed) lines. The rate of "stuff" flowing through those regions is $\vec{J} \cdot \vec{A_B} = \vec{J} \cdot \hat{k} A_B$ and $\vec{J} \cdot \vec{A_L} = \vec{J} \cdot \hat{k} A_L$.

If there are no sources or sinks that create or destroy stuff inside a small box surrounding a point, then the change in the amount of stuff in the volume of the box must be related to some integral over the box's surface:

$$\frac{d}{dt}(\text{amount of stuff in box}) = \frac{d}{dt}\int_{\text{box}}(\frac{\text{amount of stuff}}{\text{volume}})dV$$

$$= \int_{\text{box}}\frac{d}{dt}(\frac{\text{amount of stuff}}{\text{volume}})dV$$

$$= \int_{\text{box}}(\text{some scalar function related to } \vec{J})dV$$

$$= \int_{\substack{\text{box} \\ \text{surface}}} \vec{J} \cdot d\vec{A}$$
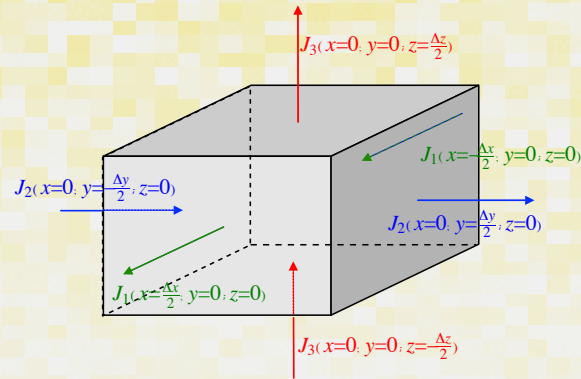
(16-1)

Figure 16-18: Integration of a vector function near a point and its relation to the change in that vector function. The rate of change of stuff is the integral of flux over the outside—and in the limit as the box size goes to zero, the rate of change of the amount of stuff is related to the sum of derivatives of the flux components at that point.

To relate the rate at which "stuff M" is flowing into a small box of volume $\delta V = dxdydz$ located at (x, y, z) due to a flux $\vec{J}$, note that the amount that M changes in a time Δt is:

$$
\begin{aligned}
\Delta M(\delta V) &= (M \text{ flowing out of } \delta V) - (M \text{ flowing in } \delta V) \\
&= \begin{array}{ll} \vec{J}(x - \frac{dx}{2})\hat{i}dydz - & \vec{J}(x + \frac{dx}{2}) \cdot \hat{i}dydz \\ + \ \vec{J}(y - \frac{dy}{2})\hat{j}dzdx - & \vec{J}(y + \frac{dy}{2}) \cdot \hat{j}dzdx \ \Delta t \\ + \ \vec{J}(z - \frac{dz}{2})\hat{k}dxdy - & \vec{J}(z + \frac{dz}{2}) \cdot \hat{k}dxdy \end{array} \\
&= -(\frac{\partial J_x}{\partial x} + \frac{\partial J_y}{\partial y} + \frac{\partial J_z}{\partial z})\delta V \Delta t + \mathcal{O}(dx^4)
\end{aligned}
$$

(16-2)

If $C(x, y, z) = M(\delta V)/\delta V$ is the concentration (i.e., stuff per volume) at (x, y, z), then in the limit of small volumes and short times:

$$\frac{\partial C}{\partial t} = -\left(\frac{\partial J_x}{\partial x} + \frac{\partial J_y}{\partial y} + \frac{\partial J_z}{\partial z}\right) = -\nabla \cdot \vec{J} = -\text{div}\vec{J} \tag{16-3}$$

For an arbitrary closed volume V bounded by an oriented surface ∂V:

$$\frac{dM}{dt} = \frac{d}{dt}\int_V C dV = \int_V \frac{\partial C}{\partial t} dV = -\int_V \nabla \cdot \vec{J} dV = -\int_{\partial V} \vec{J} \cdot d\vec{A} \tag{16-4}$$

The last equality

$$\int_V \nabla \cdot \vec{J} dV = \int_{\partial V} \vec{J} \cdot d\vec{A} \tag{16-5}$$

is called the Gauss or the divergence theorem.

London Dispersion Potential due to a Finite Body

If the London interaction (i.e., energy between two induced dipoles) can be treated as a $1/r^6$ potential, then the potential due to a volume is an integration over each point in the volume and and arbitrary point in space. This calculation will be made much more efficient by turning the volume integral into a surface integral by using the divergence theorem.

Numerical integration is a cpu-time-consuming numerical procedure. If there is a way to reduce the dimensionality of the integration, then we can reap rewards for our cleverness. One trick is to use the divergence theorem to push the integration over a volume, to an integration over a surface. For example, we could use the divergence theorem:

$$\iiint_{volume} \nabla \cdot \vec{P} \, dV = \iint_{surface} \vec{P} \cdot d\vec{A}$$

For a $1/r^6$ potential, we must find a vector potential $\vec{P}$ such that $\nabla \cdot \vec{P} = -1/|\vec{r} - \vec{x}|^6$ where $\vec{r}$ is a position in the integrated volume and $\vec{x}$ is a point at which the potential is measured.

```
PVecLondon =
          1
  ─────────────────────────────────              1
  3 ((CX - X)² + (CY - Y)² + (CZ - Z)²)³

  {CX - X, CY - Y, CZ - Z}
```

```
Needs["VectorAnalysis`"]                          2
```

```
FullSimplify[                                     3
  Div[PVecLondon, Cartesian[CX, CY, CZ]]]
```

We will integrate over a cylinder of radius R and length L along the z-axis, with its middle at the origin. First, let's use the radius of the cylinder to scale all the length variables: Let (X,Y,Z)/R = (x,y,z); (CX,CY,CZ)/R = (cx,cy,cz), and L/R = λ (the cylinder's aspect ratio).

```
ScaleRules = {X → x R, Y → y R, Z → z R,
  CX → cx R, CY → cy R, CZ → cz R};          4
PvecR5 = FullSimplify[R^5 PVecLondon /.
  ScaleRules, Assumptions -> R > 0]
```

Therefore, $\phi(\vec{x}) = \iiint_{volume} \frac{-1}{(\vec{r} - \vec{x})^6} \, dV = \iint_{surface} \overline{PVecLondon} \cdot d\vec{A} =$

$(\iint_{cylinder\ surface} \overline{PVecR5} \cdot d\vec{A} + \iint_{cylinder\ ends} \overline{PVecR5} \cdot d\vec{A})/R^5$
is the total interaction between a point and a cylinder. We can exploit the symmetry of the cylinder: $\rho = \sqrt{x^2 + y^2}$ and z.

We will do three integrals over the cylindrical surfaces using this expression to define the cylinder: (cx,cy,cz) = (Cos[θ], Sin[θ], cz):
The cylindrical surface is the domain $\theta \in (0, 2\pi)$, $cz \in (-\frac{\lambda}{2}, \frac{\lambda}{2})$

The two caps $r \in (0,1)$, $\theta \in (0, 2\pi)$, $cz = \pm\frac{\lambda}{2}$

1: To find a vector potential, $\vec{F}$ which has a divergence that is equal to $\nabla \cdot \vec{F} = -1/\|\vec{X} - \vec{CX}\|^6$, *PVecLondon* is a 'guess.' The $\vec{CX}$ will vary over the solid body and $\vec{X}$ is an arbitrary point at which the potential is to be determined.

2: We will need `Div` from the `VectorAnalysis` package.

3: this will show that the guess *PVecLondon* is a correct vector function for the $-1/r^6$ potential.

4: Our calculation will be for a cylinder of radius R and aspect ratio $\lambda \equiv L/R$. We will use R to scale all length variables and introduce dimensionless variables: $x = X/R$, $y = Y/R$, $z = Z/R$, $cx = CX/R$, $cy = CY/R$, and $cz = CZ/R$. The variables are scaled by introducing *ScaleRules* which are rules to be used in a replacement. Because the potential has a $1/(\text{length}^5)$ length scale, we multiply it by R^5 to remove that dimension. We use `FullSimplify` after the replacement with `Assumptions` of a positive radius to find the simplest possible form of the non-dimensionlized vector potential.

Cylinder Surface and Integrands

We parameterize the cylinder surface and compute the local oriented surface area and then find the integrand which is to be used for the cylinder surface.

The following is a parametric representation of a cylinder surface that is coaxial with the z-axis (the cylinder ends will be included later)

```
CylSurf = { Cos[θ], Sin[θ], cz}
```
1

The infinitessimal surface vectors $\vec{R_u}$ and $\vec{R_v}$ for the cylinder surface are obtained by differentiation; they will be used to find the surface patch $d\vec{A}$.

```
CylSurfRθ = D[CylSurf, θ]
CylSurfRcz = D[CylSurf, cz]
```
2

The surface normal given by $\vec{R_u} \times \vec{R_v}$ for the cylinder surface, there for the following (multiplied by dθ dz) is the infinitessimal oriented surface patch $d\vec{A}$.

```
NormalVecCylSurf =
  Cross[CylSurfRθ, CylSurfRcz]
```
3

The integrand to be evaluated over the cylinder surface is the vector potential, dotted into the normal vector. Because of the cylindrical symmetry of this model, we can convert to cylindrical coordinates. One set of coordinates is for the cylinder surface ($\xi \to R \, Cos[\theta], \eta \to R \, Sin[\theta]$) for fixed radius R (which is a model parameter) and another set of coordinates for where we will be testing the potential ($x \to \rho \, Cos[\alpha], y \to \rho \, Sin[\alpha]$). Because the potential must be independent of α, we might as well set it to zero.

```
CylinderIntegranddθdζ = FullSimplify[
  (PvecR5 /. {cx → Cos[θ], cy → Sin[θ],
    x → ρ , y → 0}).NormalVecCylSurf]
```
4

We have a choice whether to integrate over cz $\in (-\frac{1}{2}, \frac{1}{2})$ or $\theta \in (0, 2\pi)$ first. If we can a closed form for the cylinder surface over cz and then the cylinder end over r, then we can integrate the sum of these over θ together.

In the next section, we will see if we can do one of the two integrals---we have a choice of integrating over θ or (ζ for the cylinder sides, and R) for the cylinder ends. We find a closed form for integrating ζ for the sides and R for the top, and then subsequently numerically integrate θ for (0,2π).

1: This is the cylinder surface in terms of cz and θ

2: These are the differential quantities that define the local tangent plane to the cylindrical surface.

3: This will be the multiplier elemental area for a parameterized cylindrical surface $d\vec{r}/d\theta \times d\vec{r}/dz$, this is the local normal to the surface; here it is the unit normal because we have scaled all length quantities by R

4: *CylinderIntegrandθζ* is the integrand (i.e., the vector potential evaluated on the parameterized cylinder surface) for the cylindrical surface. Because of the cylindrical symmetry of the potential, the potential must be depend only on the normalized distance from the cylinder axis, ρ, and the height above the mid-plane, z: this conversion to cylindrical coordinates is effected by a rule-replacement operation.

3.016 Home

Full Screen

Close

Quit

MIT
3.016

Integrating over the Cylinder Surface

We have a choice whether to integrate over $\theta \in (0, 2\pi)$ or over $z \in -\lambda/2, \lambda/2$ first. We calculate the integral over cz first which will leave a form that we can numerically integrate over θ. (Note: As of 23 Oct. 2007, I've determined that *it is possible* to find the definite integral over θ and then over cz; therefore, this integral does has a closed form solution. For purposes of this demonstration, we will leave the integral over θ to be computed by a numerical integration. To demonstrate the idea of reducing the triple numerical integration, over a single numerical integration, I'll have to find a more complicated surface to integrate over in the future.)

```
UpperPlane = {λ > 0, ρ > 0, z > 0, 0 < θ < 2π};
CylinderIntegrandUpperZdθ =
 FullSimplify[Integrate[
   CylinderIntegranddθdξ, {cz, -λ/2, λ/2},
   Assumptions → UpperPlane],
  Assumptions → UpperPlane]
```
1

Here we restrict z to the upper half-space. We will treat z=0 below.

Here is the limit of the integral for z> 0 (CylinderIntegranddθ) in the limit as z → 0.

```
CylinderIntegranddθZeroLimit = FullSimplify[
  Limit[CylinderIntegrandUpperZdθ, z → 0]]
```
2

Here is the limit of the integral z=0, it is not obvious that the limit and its value at z=0 are the same.

```
FewerAssumptions =
  {R > 0, λ > 0, ρ > 0, 0 < θ < 2π};
CylinderIntegrandAtZerodθ = Integrate[
  Evaluate[CylinderIntegranddθdξ /. z → 0],
  {cz, -λ/2, λ/2},
  Assumptions → FewerAssumptions]
```
3

The limit as z -> 0 and the integrand at z = 0 are the same, so we can use a single integrand

```
CylinderIntegranddθ[
  dist_, height_, AspectRat_] :=
 Evaluate[CylinderIntegrandUpperZdθ /.
   {ρ → dist, z → height, λ → AspectRat}]
? CylinderIntegranddθ
```
4

```
CylinderContribution[
  dist_, height_, AspectRat_] :=
 NIntegrate[CylinderIntegranddθ[dist,
   height, AspectRat], {θ, 0, 2π}]
```
5

1: Because of the mirror symmetry of the function about the $z = 0$ plane, we can restrict the integral to $z > 0$ and use this as an assumption to aid the definite integral over cz. (Note this is a time-consuming integral and simplification, in the notebook form distributed with these notes, there is a dialogue that allows the user to download a precomputed result.)

2: To determine whether we can use this integrand at the mid-plane ($z = 0$), we check to see if the limit as $z \to 0$ is the same as evaluating the integrand at $z = 0$ first, and then finding the integral that applies for $z = 0$. Here, we check the limit.

3: Here, we set $z = 0$ and integrate.

4: The limit and the case of $z = 0$ are the same, so we use the form of the integrand, *CylinderIntegrandUpperZdθ* , calculated above. We turn the expression into a function by using `Evaluate` after the rule-replacement. This method of subverting the delayed evaluation, (:=), will work *so long as the function's variables have not been assigned.* These methods will be discussed in a section below . In practice, it is probably safer to replace variables with temporary, undefined, symbols and then cut-and-paste. (It is difficult to demonstrate the cut-and-paste with static notes like these.)

5: The function defined above, *CylinderIntegranddθ* , is used as the argument to the numerical integration, `NIntegrate`, over $\theta \in (0, 2\pi)$. The produces a function, *CylinderContribution* , that gives the contribution by integrating over the cylinder surface.

3.016 Home

⏮ ◀ ▶ ⏭

Full Screen

Close

Quit

MIT
3.016

Integrating over the Cylinder's Top Surface

We parameterize the cylinder's top end-cap in terms of r (dimensionless $r < 1$) and θ, and then find a closed-form solution for the double integral over the top surface.

1
```
TopSurf = {r Cos[θ], r Sin[θ], λ/2}
```

2
```
TopSurfRθ = D[TopSurf, θ]
TopSurfRr  = D[TopSurf, r]
```

3
```
NormalVecTopSurf =
 FullSimplify[Cross[TopSurfRr, TopSurfRθ]]
```

4
```
EndAssumptions =
  {λ > 0, ρ > 0 , z > 0, -1 ≤ Cos[θ] < 1};
TopIntegranddθdr = FullSimplify[
  (PvecR5 /. {cx → r Cos[θ], cy → r Sin[θ], cz →
     λ/2, x → ρ, y → 0}).NormalVecTopSurf,
  Assumptions → EndAssumptions]
```

5
```
InsideAbovedr = Integrate[TopIntegranddθdr,
  {θ, 0, 2 π}, Assumptions →
    {0 < r < 1, λ > 0, ρ < 1 , z > λ/2}];
InsideBelowdr = Integrate[TopIntegranddθdr,
  {θ, 0, 2 π}, Assumptions →
    {0 < r < 1, λ > 0, ρ < 1 , z < λ/2}];
OutsideAbovedr = Integrate[TopIntegranddθdr,
  {θ, 0, 2 π}, Assumptions →
    {0 < r < 1, λ > 0, ρ > 1 , z > λ/2}];
OutsideBelowdr = Integrate[TopIntegranddθdr,
  {θ, 0, 2 π}, Assumptions →
    {0 < r < 1, λ > 0, ρ > 1 , z < λ/2}];
Grid[{{InsideAbovedr, InsideBelowdr},
  {OutsideAbovedr, OutsideBelowdr}}]
```

6
```
TopIntegranddr = ©InsideAbovedr
```

7
```
TopPart = Integrate[TopIntegranddr, {r, 0, 1},
  Assumptions → {λ > 0, ρ > 0 , z > 0, z ≠ λ/2}]
```

8
```
TopContribution[dist_, height_,
  AspectRat_] := Evaluate[TopPart /.
  {ρ → dist,  z → height, λ → AspectRat}]
? TopContribution
```

1−4: As in the case for the cylinder's curved surface, the top surface is parameterized, then the local tangent is computed, and the local oriented surface differential element is computed. The integrand is produced with the inner-product with the vector potential evaluated at the cylinder's top.

5−6: There is a singularity at the cylinder surface that produces a little extra work on our part to ensure that we don't evaluate at this singularity. To get a closed form of the integral over θ, it is useful to divide space into four regions where the potential is to be measured: 1) Inside the cylinder radius and above the cylinder top; 2) Inside the cylinder radius and below the cylinder top; 3) Outside the cylinder radius and above the cylinder top; 4) Outside the cylinder radius and below the cylinder top. These give the same result, so long as we don't evaluate at the cylinder's surface.

7: The top integrand in r can be integrated for $r \in (0, 1)$ and produces a closed form.

8: A function for the contribution of the upper disk, *TopContribution* , is defined.

3.016 Home

⏮ ◀ ▶ ⏭

Full Screen

Close

Quit

©W. Craig Carter

Integrating over the Cylinder's Bottom Surface

We parameterize the cylinder's bottom end-cap ($cz = -\lambda/2$) in terms of r (dimensionless $r < 1$) and θ, and then find a closed-form solution for the double integral over the bottom surface.

```
BotSurf = {r Cos[θ], r Sin[θ], -λ / 2}
```
1

```
BotSurfRθ = D[BotSurf, θ]
BotSurfRr  = D[BotSurf, r]
```
2

```
NormalVecBotSurf =
 FullSimplify[Cross[BotSurfRθ, BotSurfRr]]
```
3

```
BotIntegranddθdr =
 FullSimplify[(PvecR5 /. {cx → r Cos[θ],
     cy → r Sin[θ], cz → -λ / 2, x → ρ, y → 0}).
   NormalVecBotSurf, Assumptions →
    EndAssumptions]
```
4

```
inside = { 0 < r < 1, λ > 0, ρ < 1 , z > 0};
outside = { 0 < r < 1, λ > 0, ρ > 1 , z > 0};
BotIntegrandInsidedr =
 Simplify[Integrate[BotIntegranddθdr,
   {θ, 0, 2 π}, Assumptions → inside],
  Assumptions → inside]
BotIntegrandOutsidedr =
 Simplify[Integrate[BotIntegranddθdr,
   {θ, 0, 2 π}, Assumptions → outside],
  Assumptions → outside ]
```
5

```
BotIntegranddr = BotIntegrandOutsidedr
```
6

```
BotPart = Integrate[BotIntegranddr, {r, 0, 1},
  Assumptions → {λ > 0, ρ > 0 , z > 0}]
```
7

```
BotContribution[dist_, height_,
  AspectRat_] := Evaluate[BotPart /.
   {ρ → dist, z → height, λ → AspectRat}]
```
8

```
LondonCylinderPotential[dist_, height_,
  AspectRat_] := CylinderContribution[
   dist, height, AspectRat] +
  TopContribution[dist, height, AspectRat] +
  BotContribution[dist, height, AspectRat]
```
9

1−4: As above for the cylinder's outside and for its top surface, the bottom disk is parameterized, then the local tangent is computed, and the local oriented surface differential element is computed. The integrand is produced with the inner-product with the vector potential evaluated at the cylinder's bottom.

5−6: Similar to our method of avoiding the singularity at the top surface To get a closed form of the bottom-disk integral over θ, space is divided into two regions where the potential is to be measured: 1) Inside the cylinder; 2) Outside the cylinder. These give the same result.

7: The bottom integrand in r can be integrated for $r \in (0,1)$ and produces a closed form.

8: A function for the contribution of the bottom disk, *BotContribution* , is defined.

9: We can produce a function to compute the potential at any point in space by summing the contributions from all three cylinder surfaces. The first function is the most expensive because it contains a numerical integration over θ.

3.016 Home

Full Screen

Close

Quit

The standard practice is to define functions in Mathematica with `:=`. However, sometimes it makes sense to evaluate the right-hand-side when the function definition is made. These are the cases where the right hand side would take a long time to evaluate—each time the function is called, the evaluation would be needed again and again. The following example illustrates a case where it makes sense to use `Evaluate` in a function definition (or, equivalently defining the function with immediate assignment `=`).

As in the use of (`=`), this can result in errors if the function's variables have been defined previously. In cases where it is desirable to create a function from an expression, it is probably safest to use rule-replacement with undefined variables, observe the result, and then use cut-and-paste to define a function with a delayed evaluation in terms of these demonstrably undefined variables.

To Evaluate or Not to Evaluate when Defining Functions

This example illustrates a case in which immediate evaluation = would be preferable to delayed evaluation :=

Let's set a baseline to check efficiency. Here we check timing to integrate something

```
Timing[Integrate[Exp[Tan[x]], {x, 0, c}]]
```
1

We check the same thing again, because Mathematica may have spent some time loading algorithms to integrate

```
Timing[Integrate[Exp[Tan[x]], {x, 0, c}]]
```
2

Here, we time how long it takes to create a function (with delayed assignment), but using Evaluate on the rhs.

```
Timing[DelayedEvaluated[c_] :=
  Evaluate[Integrate[Exp[Tan[x]], {x, 0, c}]]]
```
3

The following is equivalent to the above (safer) definition---and will work so long as c is not assigned to an expression.

```
Timing[Immediate[c_] =
  Integrate[Exp[Tan[x]], {x, 0, c}]]
```
4

*The following should take the *least* amount of time to perform, but as we shall see is not as efficient in the long run.*

```
Timing[FunctionDef[c_] :=
  Integrate[Exp[Tan[x]], {x, 0, c}]]
```
5

```
? DelayedEvaluated
? Immediate
? FunctionDef
```
6

The following should give a rapid result

```
Timing[DelayedEvaluated[0.5]]
Timing[Immediate[0.5]]
```
7

The following will not be rapid, because it has to do the symbolic integration before returning the result.

```
Timing[FunctionDef[0.5]]
```
8

1: When a non-trivial integral is done for the first time, Mathematica loads various libraries. Notice the difference in timing between this first computation of $\int \exp[\tan(x)]dx$ and the following one.

2: The second evaluation is faster. Now, a baseline time has been established for evaluating this integral symbolically.

3: Here, to make a function definition for the integral, the symbolic integral is obtained and so the function definition *takes longer* than it would if we had not used `Evaluate`.

4: Using an = is roughly equivalent to using `Evaluate` above and the time to make the function assignment should be approximately the same.

5: Here, the symbolic integration is delayed until the function is called (later). Therefore, the function assignment is very rapid.

6: We can use the `?`-operator to investigate the stored forms of the three function definitions. The first two forms are roughly equivalent, except for the delayed versus immediate function definition. The third form uses the unevaluated integral in the definition. symbolic information.

7: The function evaluation is much faster in the case where the symbolic integration is not needed. This would be the preferred form if the function were to be called many times.

8: The relatively slow speed of the function which contains the unevaluated integral indicates that it would be a poor choice when numerical efficiency is an issue. Therefore, if we were to use `ContourPlot`, or some other function that would need to compute the result at many different points, then the integration would be done at each point, instead of having its closed form evaluated. Thus, the function with the embedded closed form is preferable.

3.016 Home

◀◀ ◀ ▶ ▶▶

Full Screen

Close

Quit

Visualizing the Hamaker Potential of a Finite Cylinder: Contours of Constant Potential

We use the function that we have defined above as the argument to `ContourPlot`. Because the function is singular at the cylinder surface, we choose to plot the logarithm of the potential instead. Because the potential is negative outside of the cylinder, we must use an absolute value before taking the log. To remind ourselves that the potential is negative outside the cylinder, we multiply the log by minus-one.

```
LogAbsCyl[ρ_, h_, AR_] :=
 Log[Abs[LondonCylinderPotential[ρ, h, AR]]]
```
1

```
Plot[LogAbsCyl[0.0, h, 4], {h, 0, 2.5},
 Exclusions → {2.0}, PlotRange → {0, 18},
 PlotStyle → {Thick, Darker[Blue]},
 BaseStyle → {Medium}]
```
2

```
Plot[LogAbsCyl[x, 0, 1],
 {x, 0.0, 1.5}, Exclusions → {1.0},
 PlotRange → {0, 20}, PlotStyle → {Thick, Red},
 BaseStyle → {Medium}, ImageSize → Large]
```
3

```
conplotouter =
 ContourPlot[-LogAbsCyl[dist, h, 4],
  {dist, 0, 1.5}, {h, 0, 2.5},
  RegionFunction → Function[{dist, h},
   dist > 1.01 || h > 2.01],
  PlotRange → {-15, 1}, Contours → 15,
  ColorFunction → "AvocadoColors",
  AspectRatio → Automatic,
  ImageSize → Medium, Exclusions →
   {{dist == 1.0, Abs[dist - 1.0] < 0.001},
    {h == 2.0, Abs[h - 2] < 0.001}}]
```
4

```
conplotinner =
 ContourPlot[LogAbsCyl[dist, h, 4]∧],
 {dist, 0, 1.5}, {h, 0, 2.5}, RegionFunction →
  Function[{dist, h}, dist < 0.99 && h < 1.99],
 PlotRange → {0, 15}, Contours → 16,
 ColorFunction → "LakeColors",
 AspectRatio → Automatic, ImageSize → Medium,
 BaseStyle -> {Medium}, AspectRatio → Automatic
```
5

```
Show[conplotinner, conplotouter]
```
6

1: We define a short-hand function to wrap around the potential function so that the log($|P|$) is computed.

2–3: To get an idea of what the function looks like, we plot the potential first along the cylinder axis, and then for a distance within the mid-plane.

4: We break the contour-plots into an inner and an outer graphic. Here we use `ContourPlot` to plot (minus) the logarithm of the potential outside the cylinder. `RegionFunction` is used to to limit the region over which the plot is computed and displayed. Furthermore, the numerical integration is ill-behaved along lines that continue from the cylinder's corner; we use `Exclusions` to avoid these regions. We use a green tone, `AvocadoColors`, to indicate the negative values.

5: Here, we produce the contour-plot for the region inside the cylinder. Again, we use the `RegionFunction`-option of `ContourPlot`. We use blue tones, `LakeColors`, to indicate the positive potential values.

6: We combine the inner and outer regions into a single plot by using `Show`.

3.016 Home

Full Screen

Close

Quit

Visualizing the Hamaker Potential of a Finite Cylinder: Three-Dimensional Plots

We produce and equivalent visualization with `Plot3D`. From the form of these plots, it is clear that a non-polar molecule would be attracted to the cylinder, with a force that becomes unbounded in the vicinity of the cylinder. The barrier to cross into the cylinder is infinite at the cylinder surface. However, within a cylinder there is a force that pushes a foreign particle to the center of the cylinder. The Hamaker force would tend to push pores towards the middle of a dielectric cylinder.

```
plotoutside3D =
 Plot3D[-LogAbsCyl[dist, h, 4], {dist, 0, 1.5},
  {h, 0, 2.5}, RegionFunction → Function[
   {dist, h}, dist > 1.01 || h > 2.01],
  PlotRange → {-15, 1}, MeshFunctions -> {#3 &},
  ColorFunction → "AvocadoColors",
  AspectRatio → Automatic,
  ImageSize → Large, BaseStyle -> {Medium},
  AspectRatio → Automatic]
```
1

```
plotinside3D =
 Plot3D[LogAbsCyl[dist, h, 4], {dist, 0, 1.5},
  {h, 0, 2.5}, RegionFunction → Function[
   {dist, h}, dist < 0.99 && h < 1.99],
  PlotRange → {0, 15}, MeshFunctions -> {#3 &},
  ColorFunction → "LakeColors",
  AspectRatio → Automatic,
  ImageSize → Large, BaseStyle -> {Medium},
  AspectRatio → Automatic]
```
2

```
Show[plotinside3D,
 plotoutside3D, PlotRange → {-15, 15}]
```
3

4
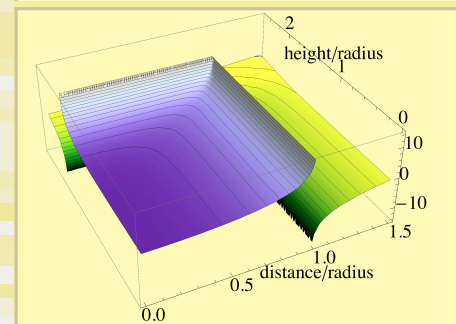
1–2: `Plot3D` is used as in the previous example with the `RegionFunction` option to separate the inner- from the outer-evaluation. The `MeshFunctions` option is used to produce shading that is consistent with the contour plots in the previous example.

3: We use `Show` with an extended `PlotRange` to produce the combined three dimensional surface representing the potential as a function of distance from the axis cylinder and height above its mid-plane.

3.016 Home

3.016

Full Screen

Close

Quit

Stokes' Theorem

The final generalization of the fundamental theorem of calculus is the relation between a vector function integrated over an oriented surface and another vector function integrated over the closed curve that bounds the surface.

A simplified version of Stokes's theorem has already been discussed—Green's theorem in the plane can be written in full vector form:

$$\int\int_R \left(\frac{\partial F_2}{\partial x} - \frac{\partial F_1}{\partial y}\right) dxdy = \int_R \nabla \times \vec{F} \cdot d\vec{A}$$
$$= \oint_{\partial R} (F_1 dx + F_2 dy) = \oint_{\partial R} \vec{F} \cdot \frac{d\vec{r}}{ds} ds$$

(16-6)

as long as the region R lies entirely in the $z = $ constant plane.

In fact, Stokes's theorem is the same as the full vector form in Eq. 16-6 with R generalized to an oriented surface embedded in three-dimensional space:

$$\int_R \nabla \times \vec{F} \cdot d\vec{A} = \oint_{\partial R} \vec{F} \cdot \frac{d\vec{r}}{ds} ds$$

(16-7)

Plausibility for the theorem can be obtained from Figures 16-15 and 16-16. The curl of the vector field summed over a surface "spills out" from the surface by an amount equal to the vector field itself integrated over the boundary of the surface. In other words, if a vector field can be specified everywhere for a *fixed* surface, then its integral should only depend on some vector function integrated over the boundary of the surface.

Maxwell's equations

The divergence theorem and Stokes's theorem are generalizations of integration that invoke the divergence and curl operations on vectors. A familiar vector field is the electromagnetic field and Maxwell's equations depend on these vector derivatives as

well:

$$\nabla \cdot \vec{B} = 0 \qquad\qquad \nabla \times \vec{E} = \frac{\partial \vec{B}}{\partial t}$$

$$\nabla \times \vec{H} = \frac{\partial \vec{D}}{\partial t} + \vec{j} \qquad\qquad \nabla \cdot \vec{D} = \rho$$

in MKS units and the total electric displacement $\vec{D}$ is related to the total polarization $\vec{P}$ and the electric field $\vec{E}$ through:

$$\vec{D} = \vec{P} + \epsilon_o \vec{E} \tag{16-9}$$

where ϵ_o is the dielectric permittivity of vacuum. The total magnetic induction $\vec{B}$ is related to the induced magnetic field $\vec{H}$ and the material magnetization through

$$\vec{B} = \mu_o(\vec{H} + \vec{M}) \tag{16-10}$$

where μ_o is the magnetic permeability of vacuum.

Ampere's Law

Ampere's law that relates the magnetic field lines that surround a static current is a macroscopic version of the (static) Maxwell equation $\nabla \times \vec{H} = \vec{j}$:

Gauss' Law

Gauss' law relates the electric field lines that exit a closed surface to the total charge contained within the volume bounded by the surface. Gauss' law is a macroscopic version of the Maxwell equation $\nabla \cdot \vec{D} = \rho$:

Index

3.016 Home

Full Screen

Close

Quit

©W. Craig Carter