MECH 222 Computer Lab 3—Parametric Surfaces and Surface Integrals

Learning Objectives

- Matlab: After reading the Briefing Notes and completing this lab, you should be able to
  - write a general function that approximates the surface integral for a given function over a 3D surface defined by a standard triangulation,
  - build a rectangular mesh of points in the plane using `linspace` and `meshgrid`,
  - triangulate a given set of points in the plane using `delaunay`,
  - triangulate some plane surfaces with curved boundaries and some flying surfaces in space by "lifting" triangulated rectangular grids,
  - generate pictures of triangulated surfaces using `triplot` and `trisurf`, and
  - [optional] estimate and exploit the order of accuracy for the approximate method described above.
- Mathematics: These readings and activities should help you learn to
  - predict integral values involving symmetry, and clearly articulate the underlying ideas,
  - translate questions about the world into a double integral or a surface integral,
  - directly experience the work of splitting a given plane domain into ribbons and fibres for the purpose of accumulating a double integral, and
  - use independent scalar parameters (u, v) to describe a flying surface in 3D.
- Engineering: This lab should help you
  - express surface area as a surface integral,
  - express a moment of inertia as a surface integral,
  - express total heat transfer in a simple situation as a surface integral,
  - compute approximate numerical values for surface integrals and double integrals, and
  - estimate the reliability of the computed approximations.

PRE-LAB ASSIGNMENT

Keep a copy of your pre-lab results with you during the in-lab activities.

PL1: Given a general surface $\mathcal{S}$ in space, suppose the areal density at each point (x, y, z) on $\mathcal{S}$ is given by $\rho(x, y, z)$ kg/m^2. Use the interpretation of surface integrals described in the Briefing Notes to answer the following questions.

(a) Find the function f_1 for which the total mass of $\mathcal{S}$ is given by the surface integral

$$J_1 = \iint_{\mathcal{S}} f_1(x, y, z)\, dS.$$

(b) Find the function f_2 for which the moment of inertia for $\mathcal{S}$ around the z-axis is given by the surface integral

$$J_2 = \iint_{\mathcal{S}} f_2(x, y, z)\, dS.$$

Hint: The moment of inertia around the z-axis for a point with mass m at location (x, y, z) is $m(x^2 + y^2)$.

(c) Use known facts from Physics/Dynamics to express the following surface integral as a function of R:

$$J = \iint_{\mathcal{S}} (x^2 + y^2)\, dS, \quad \text{where } \mathcal{S} = \left\{(x, y, z) \,:\, x^2 + y^2 + z^2 = R^2\right\}.$$

One relevant fact is that the moment of inertia for a uniform spherical shell of radius R and mass M around an axis through its centre is $I = \frac{2}{3}MR^2$. (It's important to get this right before the lab, because the exact value is needed in Activities 1 and 2. So check: (1) The units of your answer must be compatible with the units in the definition of J, and (2) since J is independent of mass M and density ρ, such quantities must not appear in your answer.)

PL2: Use mathematical notation to write down the surface integral described in test 1.2 of Activity 1. The exact value of this integral is "obvious": say what the value is, and confirm the two tests needed to justify your answer. English words are welcome—use up to 20.

PL3: Sketch the plane region D defined in Activity 4, below, and find the exact value of

$$I = \iint_D 2xy\, dA.$$

(*Check*: An approximate value is $I \approx 2.8 \times 10^2$.)

PL4: Find the function f for which the total thermal energy gain described in Activity 4.2 can be expressed as the surface integral

$$\iint_{\mathcal{S}} f(x, y, z)\, dS.$$

Take guidance from the interpretation of surface integrals described in the Briefing Notes and the units provided in the text of Activity 4.2.

ACTIVITY 1: *Write and test a surface-integral function.*

Write a new function named `surfint` so that the command

```
J = surfint(FACELIST,X,Y,Z,F)
```

assigns to the variable `J` a computed approximation to

$$J = \iint_{\mathcal{S}} f(x, y, z)\, dS. \tag{1}$$

The input arguments `FACELIST`, `X`, `Y`, `Z` describe a triangulated surface in 3D that approximates $\mathcal{S}$, and the input vector `F` supplies the (scalar) value of $f(x, y, z)$ at each vertex. In a mixture of math and Matlab notations,

$$\texttt{F(i)} = f(\texttt{X(i)}, \texttt{Y(i)}, \texttt{Z(i)}).$$

Head Start. The Briefing Notes explain how the desired calculation is a pretty straightforward extension of the surface-area calculation you did in Computer Lab 2. Your instructor's version of a surface-area calculating script and its supporting triangle-analysis function are available on Vista in the files `findarea.m` and `n_dS.m`. Feel free to base your work in this activity either on your own code from Computer Lab 2, or on these given elements.

Surface Area. Given only the elements of a standard triangulated surface, saying `F = ones(size(X))` will realize the constant function $f \equiv 1$. With this definition, `surfint(FACELIST,X,Y,Z,F)` should return the triangulated surface's area.

Testing. Try three instances of (1) involving the unit sphere,

$$\mathcal{S}: \qquad x^2 + y^2 + z^2 = 1. \tag{2}$$

Get a copy of the function `makesphere` from the lab's web page. It returns a triangulated approximation for the sphere $x^2 + y^2 + z^2 = R^2$ involving $2N^2$ triangles if you say

```
[FACELIST,X,Y,Z] = makesphere(R,N);
```

For a quick visual check that this command worked, say `trisurf(FACELIST,X,Y,Z)`.

1.1. Take $f \equiv 1$. Apply your function `surfint` to the extremely coarse approximations for the sphere returned by `makesphere` when `N=4` and `5`. You should get about 9.43 and 10.47.

1.2. Take $f(x, y, z) = xe^z$. Apply your function `surfint` to the same triangulations used in 1.1. You should get about -0.06 and 0.02.

1.3. Use `surfint` with $\mathtt{N} = \mathtt{20}$ to find an approximate value for

$$J = \iint_{\mathcal{S}} (x^2 + y^2)\, dS.$$

Find the percentage error in the approximation. (Use the exact result from PreLab PL1.)

Hand-in Checklist:

- ☐ Your function `surfint` and any other supporting functions not built into Matlab. [Remember our standard rules: every segment of code you produce and every plot you draw must include your name and UBC ID number *in electronic form*.]
- ☐ A copy of the commands you entered to make `surfint` do the work in tests 1, 2, and 3 above. **Report at least 4 digits after the decimal point in each calculated value.**
- ☐ The percentage error in test 3, supported with formulas showing how you found it.

ACTIVITY 2: *Triangulate a plane rectangle and a surface above it.*

Make a triangulation of the plane rectangle

$$D = \left[0, \frac{\pi}{2}\right] \times \left[0, \pi\right]$$

in the style of Figure 1 in the briefing notes, but use N dots in the x-direction and M dots in the y-direction, with (M, N) defined by your assigned lab day:

Mon: (5,9), Tue: (7,12), Wed: (10,6), Fri: (4,8).

Then use this triangulation to produce a triangulated approximation for the surface

$$\mathcal{S}: \qquad z = 2\sin(x)\sin(y), \qquad (x, y) \in D.$$

Produce a 3D plot similar to Figure 2 in the briefing notes, showing both triangulations on the same axes. Adjust the viewing angle and use Matlab to label the axes and put your name in the plot title. The mesh looks best if you can see between its edges. The options shown in the model command below help with this:

```
trimesh(FACELIST,X,Y,Z,'FaceAlpha',0);
```

Hand-in Checklist:

- ☐ The plot specified above.
- ☐ A transcript of the commands you used to make that plot. (If you wrote a script, print that; if you just typed commands one at a time, copy them from the command window into some text document for printing.)

ACTIVITY 3: *Triangulate a more general plane region and a surface above it.*

Grid up a rectangle in the (u, v)-parameter plane using 11 nodes along each coordinate axis. Use this to build and plot a triangulation of the plane set

$$D = \left\{(x, y) \,:\, y^2 - 4 \le x \le 3y\right\}.$$

(You should have a handmade sketch of D from PreLab PL3.) Then use this triangulation to produce a triangulated approximation for the surface

$$\mathcal{S}: \qquad z = \frac{1}{20}(1+y)(3+x)(12-x) + \frac{2}{3}(3y - x), \qquad (x, y) \in D. \tag{4}$$

Produce a single 3D plot showing your triangulations of D in the plane $z = 0$ and the triangulated surface S flying above it. (It should look a little like a kitesurfer's kite.) Adjust the viewing angle and use Matlab to label the axes and put your name in the plot title.

Hand-in Checklist:

- ☐ The plot specified above.
- ☐ A transcript of the commands you used to make that plot.

ACTIVITY 4: *Approximate two integrals.*

4.1. Pretend you don't know the exact value of the integral I in PreLab question PL3. Show how you could use triangulation and `surfint` to estimate I to the nearest integer multiple of 0.01. Briefly explain why you trust your estimate, making reference *only* to evidence acquired from computations. (In most practical situations, the exact result is not available for comparison!)

4.2. Suppose all coordinates are measured in metres, and the surface $\mathcal{S}$ defined in (4) is made of a space-age fabric designed to collect heat. At dawn, the surface is at a uniform temperature of 6°C. At noon, its temperature (°C) at (x, y, z) is given by $T(x, y, z) = 10+2z$. Find the amount of thermal energy gained by the surface between dawn and noon. (For temperatures in the range of interest, the fabric's heat capacity is 2093 J/(m^{2}K).) Get at least 3 digits correct. Note that the grid used for sketching $\mathcal{S}$ in Activity 3 is probably too coarse to achieve the desired accuracy.

Hand-in Checklist: For each integral, supply

- ☐ Mathematical notation for the surface integral to be found.
- ☐ The calculated numerical value.
- ☐ A transcript of the commands you used to find the value.
- ☐ A terse explanation (one or two lines) of why the value is trustworthy.

BONUS ACTIVITY 5 (OPTIONAL): *Estimate the order of accuracy for* `surfint`.

Use numerical experiments to find p, the integer exponent in the error estimate

$$E(h) \stackrel{\text{def}}{=} J(h) - J(0) \approx Ch^p, \qquad 0 < h \ll 1. \tag{5}$$

Recall that $J(h)$ denotes the approximate value returned by using `surfint` and `makesphere` with characteristic triangle length $h = 1/N$, and $J(0)$ is defined to be the exact theoretical value of the integral. Work with the moment of inertia integral discussed in prelab PL1, taking $R = 1.5$. That is,

$$J(0) \stackrel{\text{def}}{=} \iint_{\mathcal{S}} (x^2 + y^2)\, dS, \qquad \mathcal{S}:\ x^2 + y^2 + z^2 = \frac{9}{4}.$$

The exact value should be known from PreLab PL1.

5.1. Calculate 5 to 10 increasingly accurate approximations $J(h)$ by increasing the node-counts in the triangulation and re-running `makesphere` and `surfint`. Consider writing a script to automate this process. Make a table of values in which each number h is shown with the calculated value $J(h)$, the absolute error $E(h)$, and the corresponding percentage error. (Reasonable choices for N depend on your need for accuracy and the speed of your machine. Case $N = 10$ quickly makes a surface that is obviously a poor approximation for a sphere, so it's too small. Case $N = 200$ takes 11.5 seconds to run on a Math Dept server. That's a long wait, but may be worth it for a single run after the code is thoroughly tested.)

5.2. Use a log-log plot to identify the most likely integer value of p in (5) above.

5.3. Once you know p, use it to make a more accurate prediction of $J(0)$ based on the two finest triangulations in your table. Find the percentage error in this prediction.

Details: Plugging a known grid spacing h_1 and computed value $J(h_1)$ into line (5), where you now know p, gives a linear equation with $J(0)$ and C as the two unknowns. Plugging in a second grid spacing and calculated value gives you a second linear equation. Now you have a 2×2 system: solve it for $J(0)$. Unfortunately the linear equations are actually just approximations, so the value you get for $J(0)$ is not exact—but it's typically closer than either of the approximations you worked hard to calculate. (This is **Richardson Extrapolation.**)

Hand-in Checklist:

- ☐ The table of values specified in 5.1.
- ☐ A plot showing $y = \log(E(h))$ as a function of $x = \log(h)$.
- ☐ Your predicted value for the order p, supported by a short explanation.
- ☐ The approximation to $J(0)$ produced by Richardson Extrapolation using the two finest computed results.
- ☐ The percentage error in the approximation just found.

Appendix: A List of Relevant Matlab Commands

This list shows all the Matlab functions used in your instructor's private solutions to Activities 1–5.

General Matlab functions.

`clear`	Clear variables and functions from memory
`cputime`	Elapsed CPU time
`disp`	Display text or array
`length`	Number of elements in a vector
`linspace`	Create a vector with linearly-spaced entries
`meshgrid`	Generate X and Y arrays for 3-D plots
`round`	Round to nearest integer
`size`	Dimensions of a vector, matrix, or array
`sprintf`	Write formatted data to string
`zeros`	Create array of all zeros

General Math functions.

`abs`	Absolute value and/or complex magnitude
`exp`	Exponential
`log`	Natural logarithm
`sin`	Sine of argument in radians

General Matlab graphics functions.

`figure`	Create figure graphics object
`gca`	Current axes handle
`get`	Query Handle Graphics object properties
`plot`	2-D line plot
`plot3`	3-D line plot
`set`	Set Handle Graphics object properties
`title`	Add title to current axes
`view`	Viewpoint specification
`xlabel`	Label x-axis
`ylabel`	Label y-axis
`zlabel`	Label z-axis

Matlab functions for triangulations.

`delaunay`	Delaunay triangulation
`trimesh`	Triangular mesh plot
`triplot`	2-D triangular plot
`trisurf`	Triangular surface plot

User-supplied functions.

`makesphere`	Triangulated approximation to a sphere [provided on Vista]
`n_dS`	Returns the normal vector $\widehat{\mathbf{n}}\,dS$ for a given triangle [from Computer Lab 2]
`surfint`	Approximation to a surface integral [students make this]