You must answer all questions. For full credit, an answer must be both correct and well-presented (clear and concise). If you feel a question is ambiguous, state any assumptions that you need to make. Also, some of the questions are "essay" questions. For those, there are many correct answers. It is more important that you provide a good argument for the answers you give, than that you give the "most correct" answer. Sometimes, we are particularly looking for your ability to make a clear and concise argument based on things you are aware of, rather than to see if you can find the best possible answer, or have seen all possible research on the topic.

Question 1: The modern paper

You were asked to read the paper

> *"Regularized Kelvinets"*
> *by Fernando de Goes and Doug James*
> *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2017)*

Based on your analysis of this paper and any of its citations that you have reviewed, but also using your general knowledge of the graphics principles in your reading list, answer the following questions:

A. The method of this paper is inspired by the fact that fundamental solutions (Green's functions) of linear elasticity have a closed form, and could be used as warping functions in space that drag along the geometry of a model being manipulated. However, instead of using the Green's functions themselves (the response of the elasticity equation to a "true" Delta function), the authors use a "regularized" variant (the response of the elasticity equation to a "regularized" Delta). Why is this so? Intuitively, what would have been the visual consequence of using non-regularized fundamental solutions for this sculpting tool?

B. Would it have been advisable to use the method described in the paper to animate the individual fingers of a character's hand, or the lips on the character's face? (Assume that the reference pose of the character has the fingers reasonably close to one another and the lips of the mouth almost touching each other). If not, can you suggest any steps one could take to mitigate this limitation?

C. Briefly compare the pros and cons of this technique, relative to *as-rigid-as-possible* shape manipulation techniques (you may use either the original work of Igarashi et al'05 from your reading list as a point of reference, or the similar work of Sorkine and Alexa'07 "As rigid as possible Shape Modeling", cited in this paper).

Question 2: Linear algebra and transformations

A. Matrix multiplication, in general, does not commute. That is, for two square matrices A and B the matrix products AB and BA are generally not equal.

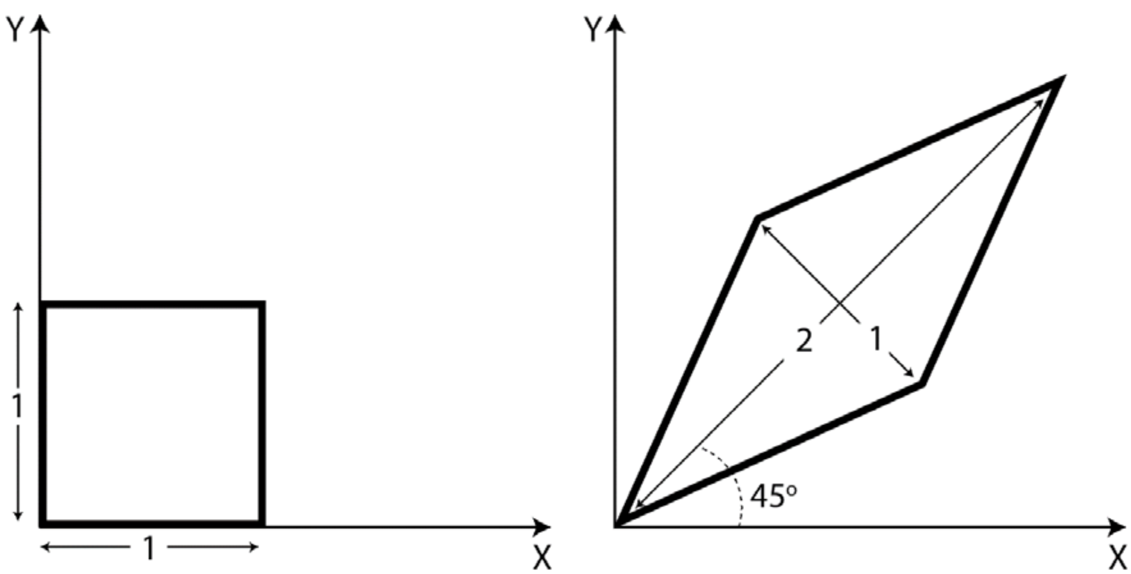
When we represent linear transformations as **matrices**, it is common to represent the result of applying two transformations in sequence by the product of their respective matrices (i.e. a non-commutative operation).

On the other hand, **rotation vectors** are often used to encode 3D rotations (also known in Graphics as exponential maps, they are also analogous to angular velocities). We may often compose rotation vectors by *adding* them, but of course vector addition does commute, in contrast to matrix multiplication.

Which of the two approaches correctly captures composition of transformations? If both of them are in fact correct and accurate, explain the apparent disparity. If one of the two approaches is not fully accurate, explain why it still does something reasonable despite being different.

B. Consider the 2D **linear transformation** (represented by a 2x2 matrix) shown in the illustration below. After applying this transformation (let's call it **T**), the unit square on the left is transformed into the rhombus on the right. Conceptually, the transformation stretched the square along one of the diagonals (the line x=y) so that this diagonal now has length 2, while the length of the other diagonal is now exactly equal to 1.

From the theory of the Singular Value Decomposition, we know that any 2x2 matrix T can be written as a product $\mathbf{T=U*D*V^T}$, where **U** and **V** are rotation matrices, and **D** is a diagonal matrix (corresponding to an axis-aligned scaling). Write down what those matrices would be in this case, and explain how this decomposition would work out intuitively. You can use notation like *Rotate(-30°)* or *Scale(3.0,0.4)* to denote rotation and scaling matrices if you like – no need to write down all their elements explicitly.



Question 3: Ray Tracing and Global Illumination

A. Imagine a bathroom with a large variety of surfaces and materials in it (including mirrors, painted walls, tiled walls, shiny fixtures, a bathtub full of water, a glass shower door, floormats, etc). Assume that there is a single **point light source** (e.g. a tiny LED light).

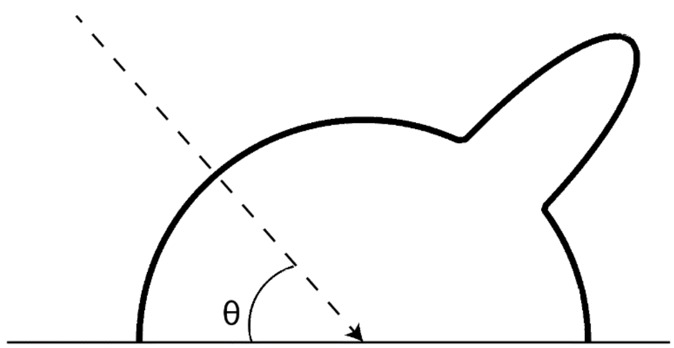
   i. Light paths are often characterized using a regular expression notation with symbols **L** (light), **D** (diffuse), **S** (specular), and **E** (eye). Describe possible scenarios in this bathroom scene that would give rise, respectively, to light paths of the following types: **LDDSE**, **LSDDE** and **LSSE** (give 1 example per path type).

   ii. The *recursive ray-tracing algorithm* initiates paths from the camera, which subsequently spawn **reflection, transmission (refraction)**, or **shadow** rays when they hit objects, on the way to light sources.
   Describe a scenario in this scene where a ray from the camera would spawn the following sequence of secondary rays (these are steps of a single light path):
   (Camera→) Transmission→Reflection→Transmission→Reflection→Shadow.

   iii. Explain why it is or is not possible to have a soft shadow given that there is only a point light source (if it is possible, give an example; if it is not possible, give an argument).

   iv. The bathroom scene we just described would probably not have been the best candidate for rendering using the classic radiosity method. Explain why, and suggest how you might restrict the types of materials in this scene to make it easier to use a radiosity renderer.

B. The figure below is a partial illustration of the **bi-directional reflectance distribution function (BRDF)** of a material with both diffuse and specular properties (it is actually very close to a Phong reflection model – ignoring ambient terms). A specific direction of the incoming light is shown (corresponding to the angle θ) and the thick contour illustrates the radiance distribution for all (outgoing) directions of reflectance.



Draw simple illustrations of what this diagram could have looked like for

   i. A purely diffuse material (use the same angle θ).
   ii. A similar Phong-type material but with with a smaller specular exponent (same θ)
   iii. An almost perfect mirror-like material (same θ).

<u>Question 4: Skinning and deformers</u>

A. Harmonic maps provide a way to implement cage-based skinning. Cage-based skinning is a recent alternative to the more straightforward approach of bone-based skinning (often implemented with linear-blend skinning, which goes by many different names; however, superior methods for implementation are mentioned in the readings).

   i. What are the pros and cons of cage-based approaches as compared to bone-based approaches?
      (Note: We are less concerned with the downsides of specific implementations, such as the candy-wrapper artifacts that occur in linear blend implementation of bone-based skinning. We are focusing on the differences of the core concepts of bone-based and cage-based techniques).

   ii. What are the features of harmonic maps that make them an attractive way to implement cage-based skinning? (i.e. why are they better than alternatives like Free-Form Deformations?)

<u>Question 5: Gradient-Domain Image Processing & Laplacian Surface Editing</u>

Gradient-domain image processing and Laplacian Surface Editing techniques hail from the distinct domains of digital image processing, and computational geometry respectively. However, the two concepts share some very fundamental commonalities in mathematical formulation and inspiration. And, of course, sometimes these methods will diverge in their theoretical approach, as a consequence of their distinct driving applications.

A. Laplacian Surface Editing, as the name reveals, has a very special role in its mathematical formulation for the Laplacian operator, and its counterpart the ***Poisson equation***. A very similar mathematical formulation shows up in Gradient-Domain image processing techniques, as well. Elaborate on the similarities between the equations showing up in both schemes (is there some intuitively similar feature or property that both methods are trying to capture?). Also, comment on the differences between the ways either method uses those seemingly similar equations.

B. One of the two methods (Gradient-Domain image processing, or Laplacian Surface Editing) pays special attention to rotation invariance, i.e. making sure that the features extracted from input data only depend on relative differences between neighboring data variables, and not on the overall position & orientation of the data itself. Rotation invariance is not really a concern for the other of the two methods. Explain which is which, and why rotation invariance is a concern for one of them but not the other.

C. One could envision a type of Geometry Editing process where the surface detail of a 3D model is represented as a displacement map, which is overlaid on a smoothed surface approximation. We could then use Gradient-Domain image processing techniques to manipulate the displacement map, and as a consequence the shape of the 3D model itself (with the modified displacement map applied). Explain why this approach is much narrower in scope and capabilities than full-fledged Laplacian Surface editing methods.

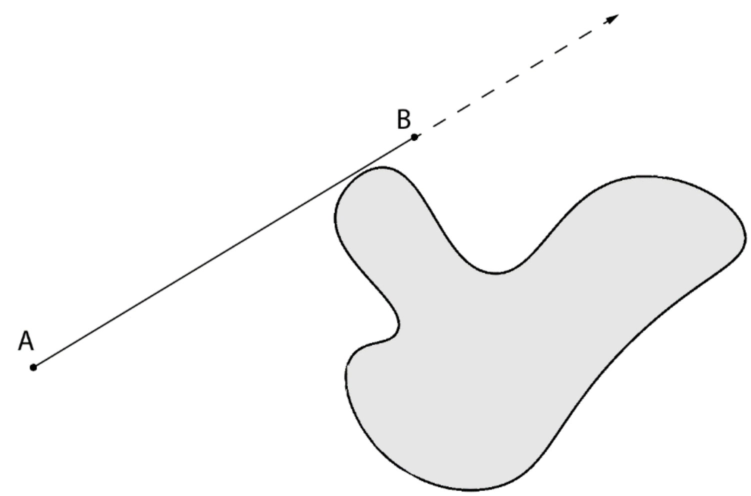
Question 6: Implicit Surfaces

Implicit surface representations define surfaces in 3D as the iso-surface associated with a certain value (typically zero) of a scalar field defined through the 3D space. Given a well-defined closed surface that we seek to represent, one popular way to construct such a scalar field is by defining a *signed distance function*. This would be a scalar field whose absolute value at a given location is equal to the distance to the closest point on the surface, while the sign is negative inside and positive outside the surface.

A. Implicit surface representations are highly convenient representations for certain models and phenomena, and wholly inappropriate for others. For the following scenarios, explain whether using an implicit surface representation would be a good option or not: (if the answer is no, suggest an alternative, e.g. meshes, point clouds, subdivision surfaces, Constructive Solid Geometry, etc)

   i. Surfaces of liquids, that may potentially be animated over time.
   ii. Items of clothing.
   iii. Storage of surface models captured using 3D scanning hardware.
   iv. Models of hair for 3D animated characters.

B. In a ray-tracer, we spawn a ray from point **A** as seen in the illustration below, and in the direction of another point **B**. A solid object (shown shaded grey) may lie in the path of the ray. Let us assume that the object's surface is described as the zero iso-surface of a signed distance function φ(x), defined through space (φ is negative inside the object, positive outside. Also |φ(x)| is equal to the distance of x from the closest point on the shaded object).



   i. Explain why, even if we had φ(A)>0 and φ(B)>0, this would not be sufficient to say with certainty that the ray segment AB doesn't intersect the object.

   ii. Show that if

$$\min(\varphi(A), \varphi(B)) > \text{length}(AB)/2$$

   this would now prove that the ray segment AB doesn't intersect the object.