The home exam was mostly checked for conceptual mistakes; I did not deduce any points if it seemed the solution is going in the right direction, but the final result did not match. However, I did so in the case where the solution was inexplicable. For the last question, I only deduced points if the solution was improbable or poorly explained.

1. Dynamics and Kinematics

1.1. Slightly-impatient birds. You are shooting a mildly-perturbed bird of no dimensions (a particle bird), and mass \( m = 2\text{[kg]} \) from a sling of height \( 1\text{[m]} \) above the ground, at point \((0, 1)\), where the up axis is \( \hat{y} \). The sling behaves like a spring of zero rest length \( l_0 = 0\text{[m]} \) with spring coefficient \( K = 0.5\text{[N/m]} \). You can calibrate both the shooting angle and the length of the spring before the release. You are shooting at a vertical wall located at \( x = 10 \), and you score higher when reaching a greater height when you hit it. What are the optimal shooting angle and length to reach the highest? The sling is of course limited by the ground.

Solution.

Let us first compute the shooting velocity, which will be the key in computing the trajectory. Ignoring the height potential energy (as instructed), we can do so by converting sling (spring) energy to kinetic energy:

\[
\frac{1}{2}m|v|^2 = \frac{1}{2}k|l|^2.
\]

The actual velocity would be \( \vec{v} = |v| (\cos \theta, \sin \theta) \), where \( \theta \) is the angle of the sling with the ground. The length of the sling is then \( l = \frac{1}{\sin \theta} \). The horizontal speed is constant, and therefore to hit a wall 10\text{m} away, it would take time \( t \) so that:

\[
t = \frac{x}{|v| \cos \theta}
\]

The vertical velocity is \( |v| \sin \theta - gt \) due to constant gravitational acceleration downwards, and the height function is then \( h = 1 + t \cdot \sin \theta - \frac{g t^2}{2} \). We substitute \( t = |v| \cos(\theta) \) and \( |v| = l \sqrt{\frac{k}{m}} = \frac{1}{\sin \theta} \sqrt{\frac{k}{m}} \) to get \( t = x \sqrt{\frac{m}{k}} \tan \theta \), and consequently:

\[
h(\theta) = 1 + x \cdot \tan \theta - \frac{gx^2 m \cdot \tan^2 \theta}{2k}
\]
deriving to get the maximum we get:

\begin{equation}
(4)\quad h'(\theta) = \frac{x}{\sin^2 \theta} - \frac{gx^2m \cdot \tan \theta}{k \cdot \sin^2 \theta}
\end{equation}

The (common) denominator is always positive (assuming the angle is never zero), and thus we need to solve:

\begin{equation}
(5)\quad h'(\theta) = 0 = x - \frac{gx^2m \cdot \tan \theta}{k \cdot \sin^2 \theta} \rightarrow \tan \theta = \frac{k^2m}{gx^2} \cdot \tan \theta = \frac{k^2m}{gx^2} \cdot \tan \theta \approx \tan(2.55 \times 10^{-3}) = 0.146^\circ.
\end{equation}

Essentially, a very low angle, which means a very high-strung sling is better than pointing up.

1.2. **Sonique the spherehog.** There is a triangular slope at coordinates (0, 0), (10, 0) and (10, 15). Sonique starts running in the bottom of a slope at a configurable constant acceleration of \[a\] [m/sec^2], leaps from the top of the slope, and immediately becomes a spherical spherehog of radius 0.25m. He naturally then stops being affected by any *external* force but gravity. There is a pile of coins at point (40, 5). What the the necessary acceleration \[a\] for sonique to exactly hit the pile of coins?

**Solution.**

We use the configurable acceleration \[a\] to express the jump velocity \[v_0\]: the distance of climbing is \[d = \sqrt{100 + 225} = 18.02\], and thus we have \[d = \frac{at^2}{2}\], giving \[t = \sqrt{\frac{2d}{a}}\] and \[v_0 = \sqrt{2at}\].

We know the values for the necessary positional difference: \[\Delta x = 40 - 10 = 30\] and \[\Delta y = 10\] (the radius cancels out since you hit the coins with the bottom; however I accepted all possible variations as answers). We have the trajectory formulas:

\begin{align}
(6)\quad \Delta x &= v_0 t \cdot \cos(\psi) \\
(7)\quad \Delta y &= v_0 t \cdot \sin(\psi) - \frac{gt^2}{2} = \Delta x \cdot \tan(\psi) - \frac{g \Delta x^2}{2v_0 \cdot \cos^2(\psi)}
\end{align}

where \(\psi\) is the angle of the slope. Putting in the numbers is a bit tedious, so we cut to the final solution which is about \[a \approx 7.25 \frac{m}{sec^2}\].

2. **Calculus**

2.1. **Maps and Transformations.**

(1) A symmetric bilinear map is represented by a matrix \(M_{n \times n}\) in some coordinate system with dimension \(n\). What is the representation in a coordinate system \(R\) (rows of \(R\) are the vectors of the system). Prove clearly.

**Solution**

Let us assume implicitly that the “natural” system where \(M\) is represented is the canonical one. (Some students did a more general job of re-representation in this system, and that’s very nice). Then, a column vector \(v\) in the canonical system is represented as \(Rv\) in the system spanned by \(R\), as it is the vector of dot product with
and every basis vector of $R$ as rows. Bilinear forms are invariant to coordinate systems, and therefore should give the same result. Thus $v^T R^T M_R R v = v^T M v$, and we consequently get that $M_R = R M R^T$.

(2) Show that if the Jacobian of a change of coordinates $f(\vec{x}) = y$ (in $n$ dimensions) is a rotation and scale, that the deformation in lengths that $f$ induces is uniform in all directions and equal at all points. What is this deformation?

Solution That doesn’t even need a very rigorous proof: we know that the new metric Bilinear form is $J_f^T \cdot J_f$. in case $J_f$ is of the form $s R$ for some scalar $s \in \mathbb{R}^+$ and $R$ a rotation matrix, then we get that the scaling of each norm of vector, or its length, is exactly by $s$, as $s R^T R s = s^2 I$ scales the squared norm.

2.2. Vector calculus.

(1) Show that a gravitational field (i.e., a gradient field of gravity) that two (immobile) stars induce in space, with masses $m_1, m_2$ and distance $d_{12}$ between them, must have at least one zero point (where the force is zero). What is the geometric set of all such zero points?

Solution it is straightforward to see that whatever this region is, it must lie in the line connecting the two stars: otherwise, the sum of forces cannot be zero, as both stars have an attraction component in the same direction (the orthogonal direction to this very line between their centers). We know that gravity is proportionate to $\frac{m_1 d_2}{d_1^2}$ and therefore this point is exactly in a position on the line where $\frac{m_1 d_1^2}{d_1^2} = \frac{m_2 d_2^2}{d_2^2}$.

While this is a quadratic equation for the distances, it in fact only has one solution, as both forces have to be in opposite directions. So we get:

$$d_1 = \sqrt{\frac{m_2}{m_1} d_2}, \quad d_1 + d_2 = d_{12}$$

where $d_{12}$ is the distance between the stars.

Bonus question of unspecified points: Next assume $n$ such stars (in general position), how many zero points must we have at maximum?

Solution. There cannot in general be more than one, and even then it’s not in general position. In essence, you are looking at a point which is (weighted) equidistant from many planets, which is like a weighted center of circle (very hand-wavingly). This doesn’t not exist unless the stars are arranged in this kind of pattern.

(2) Show that harmonic functions ($f$ is harmonic if $\Delta f = 0$. consequently, $\nabla f$ is both curl and divergence free) do not have maxima or minima in the open space $\mathbb{R}^3$.

Solution. The formal way is to look at the second derivatives $\frac{\partial^2 f}{\partial x^2}$ and understand that not all can be negative or all positive, and therefore deduce that there are directions that are always going down and directions that are going up in the function. It is also the meaning of the “mean-value theorem” where the Laplacian of a function is the average (in the continuous case, integration over) the neighbors, so you cannot be strictly more or less
than any point on a ball of any radius around the point, and therefore not a maximum or a minimum. However, answers that say that the gradient is a constant or $\nabla f = 0$ are wrong.

3. **Collisions**

Describe an algorithm to (relatively) safely avoid penetrations, and any inexact resulting resolution. Next describe the reason for your method to *not* be used in a game engine (or else if you think it’s a very good one for real-time, explain why).

**Solution.**

There were several nice answers. The most common one was to bisect within the time frame, and find the correct intersection, which is of course not feasible and especially in cases of multiple objects. One nice solution that repeated a few times is to predict the intersections by some vector or distance field in advance. This is actually being done in some painstaking way in many position-based dynamics methods.