# MATH 762 PROJECT: TRACKING SEDIMENT TRANSPORT THROUGH RIVERS WITH VARIED GEOMETRY

COLLIN KOFROTH

Abstract. In this project, we study the transport of sediment in the presence of a turbulent river flow. After using Fusion 360 to design various different fluid domains, we utilize COMSOL to solve the $k$-$\varepsilon$ turbulence model, using particle tracers to track in-flow sediment. Geometries include a 2D section of a winding river and a winding river with obstacles, to simulate the scenario of river rapids. The results of these simulations intimate that, in the cases shown, the effect of river rapids on the dispersion and deposition of sediment was negigible. Simulations with obstacles, despite minor fluctuations, settle into similar velocity, vorticity, and particle trajectory profiles.

Contents



## 1. Introduction

The transportation of sediment in bodies of water, such as rivers, can lead to erosion, contaminant transport (due to adjacent industrial activity), and negatively impact the ecosystem of associated organisms ([2]). A lack of transport and resulting sediment deposition can cause a deficiency in essential nutrients, an absence of viable cover from predators, and erosion. Conversely, too much transport and deposition can lead to the covering of underwater habitats, waterway deformation, light prevention, and the clogging of fish gills (see [4] for more details on all of the above).

Thus, the analysis of sediment transport is of great importance and interest. Obtaining reliable numerical results can be extremely difficult. The varied, rough geometry of a river typically induces a turbulent flow, especially in sharply-angled, winding rivers or rapids. Coupling flow and erosion is an additional difficulty; see [3]. Most study in the mathematical community is based on developing numerical techniques or river models (e.g. [1]-[3]), whereas we will use COMSOL, which is based on finite element methods, to solve our problem on specified geometry with the proper type of physics and will discuss quantitative results. For our purposes, we will be neglecting bed evolution, instead studying the general fluid flow and using particle tracers to understand the dispersion of in-flow sediment under the flow's evolution. This will be done for various types of geometries. The original goal was to perform 3D simulations for a winding river with a varying bed profile. This goal proved unattainable in the time required, as will be discussed. Consequently, we considered 2D flow for a winding river, as well as a winding river with obstacles, which we used to model river rapids. Here, the goal became to contrast the case of no obstacles versus obstacles. We will begin by discussing the 3D problem and its pitfalls before proceeding to the 2D results.

1.1. **Preliminaries.** In this section, we will discuss everything from the 3D perspective. We will be concerned with solving for the flow, which is typically modeled via the incompressible Navier-Stokes equations. That is, we take $u : \bar{\Omega} \times [0,\infty) \to \mathbb{R}^3,\ p : \bar{\Omega} \to \mathbb{R}$ to solve

$$\begin{cases} \rho(\partial_t u + (u \cdot \nabla)u) - \nu \Delta u + \nabla p = f, \\ \nabla \cdot u = 0, \end{cases},$$

subject to an appropriate initial condition and boundary conditions (commonly, one uses the no-slip boundary condition, $u\big|_{\partial\Omega} = 0$), where $\Omega \subset \mathbb{R}^3$ is open and bounded (say with, at least, Lipschitz boundary), $\rho$ is fluid density, $u$ is velocity, $p$ is pressure, $\nu$ is kinematic viscosity, and $f$ is the volume force.

It is worth noting that many river models, instead, use the shallow water equations. In such a scenario, the vertical length scale is significantly smaller than the horizontal length scale. Conservation of mass implies that the vertical velocity must scale negligibly, justifying a vertical hydrostatic approximation and analysis of a different set of equations in 2 space dimensions. While this is also the situation in our model, COMSOL does not directly implement the shallow water equations (one can use the general PDE interface, but we prefer to use the built-in models, as they are more robust). However, this line of thinking provides some justification for considering 2D models.

The turbulence is characterized by the Reynolds number, which is a ratio of inertial forces to external forces:

$$\mathrm{Re} = \frac{\rho U L}{\mu} = \frac{uL}{\nu},$$

where $\rho$ is the density of the fluid, $U$ is the characteristic velocity of the flow (relative to the boundary layer), $L$ is the characteristic length scale of flow, $\mu$ is the dynamic viscosity of the fluid. Large Reynolds numbers correspond to turbulent regimes, which leads to various instabilities in the flow; what "large" means is dependent on the context. In fact, the Navier-Stokes equations exhibit both hyperbolic and parabolic behavior, and the size of the Reynolds number can dictate which regime is more prevalent.

COMSOL offers a variety of turbulence models for incompressible flows and compressible, low Mach number flows (Ma $< 0.3$). Each turbulence model is based on the Reynolds-averaged Navier Stokes equations (RANS), which averages the velocity and pressure fields in time. If we let $U$ and $P$ denote time-averaged velocity and pressure, then the RANS equations are

$$\rho((U \cdot \nabla)U) + (\nu_T - \nu)\Delta U + \nabla P = f,$$

where $\nu_T$ is the turbulence viscosity. The turbulence model that we will choose is called the $k$-$\varepsilon$ turbulence model, one of the most popular in CFD. This model solves for the variables $k$ and $\varepsilon$, where $k$ is the turbulence kinetic energy and $\varepsilon$ is its rate of dissipation. It is notable for possessing a strong convergence rate and inexpensive memory demands.

These equations take the form

$$\begin{aligned}
\rho(\partial_t k + (u \cdot \nabla)k) - \left(\nu + \frac{\nu_t}{\sigma_k}\right) \Delta k &= p_k - \rho\varepsilon \\
\rho(\partial_t \varepsilon + (u \cdot \nabla)\varepsilon) - \left(\nu + \frac{\nu_t}{\sigma_\varepsilon}\right) \Delta k &= C_{\varepsilon,1}\frac{\varepsilon}{k}p_k - C_{\varepsilon,2}\rho\frac{\varepsilon^2}{k}, \\
\nu_t &= \rho C_\nu \frac{k^2}{\varepsilon}
\end{aligned}$$

where $\sigma_k, \sigma_\varepsilon, C_{\varepsilon,1}, C_{\varepsilon,2}, C_\nu$ are adjustable constants, and $p_k = 2\nu_t \mathrm{Tr}\ S^T S$, and $S = \nabla u$ is the strain-rate tensor.

## 2. Methods

2.1. **Domain Construction.** We will discuss the creation of 3D geometries, since domains in 2D were simply obtained by projecting onto the $xy$ plane. All geometries were designed in Fusion 360. We will provide the workflow for our working problem, a section of hypothetical river that features horizontal winding with a varying bed profile. The section is 200 meters in the $y$ direction, 30 meters in the $x$ direction, and the height varies, but 0.5-1 meters deep to start, with up to an additional $[0.5, 1] \ni \tilde{z}$ meters of variance in the negative $z$ direction. Our process follows this procedure:

**1. Create Horizontal Cross-Section:** First, we create a sketch in the $xy$-plane. Using a control-point spline, we create a curve with the winding that we desire that is 200 meters in length. Then, we translate this curve in the $x$ direction by 30 meters while keeping the original copy, guaranteeing that the width is 30 meters throughout. Next, we attach the corresponding endpoints, closing the sketch. From here, we extrude this planar object, and the height does not matter significantly (as long as it is a few meters high, so one does not need to worry about it being too high).

**2. Create Bed Profile:** Once again, we create a sketch in the $yz$-plane. There are two ways that we might do this. The first one: We use a control-point spline to generate a curve with the features we want, also 200 meters in length (in the $y$ direction). Switching over to the patch environment, we extrude this curve to obtain a curved surface (it is important to make sure that the distance that one extrudes is larger that 30 meters, as will be seen shortly). The second option: We do the same, then we add straight line with the same length in the $y$ direction, but 1 meter above. We connect the corresponding endpoints with lines, then we extrude this. Regardless of the option, the spline should have extremely minor variance in comparison to the $z$ scale, making it somewhat difficult to obtain a nice profile, if one draws the entire curve at once. Instead, we draw for only a fraction of the 200 meters required, then (keeping the old copies) translate the curve until we get to 200 meters. Then, we proceed as above.

**3. Splitting:** Now that we have our surfaces, we move the curved surface or extruded surface sufficiently in the positive $z$ direction (so that the top half of the split body has the appropriate depth average), then use the split body command to split our first object. This gives us our domain.

Here is a more precise workflow for a particular example, with accompanying figures in the "Figures" section. Figure 1 depicts steps 1-4, figure 2 shows steps 5-10, and figure 3 for steps 11-13.

(1) Create a sketch in the $xy$-plane
(2) Draw control point spline, with distance 200 in the $y$ direction.
(3) Keeping the old copy, translate in the $x$ direction by 30 meters, then connect the corresponding endpoints.
(4) Extrude the resulting sketch 2 meters in the $z$ direction.
(5) Create a new sketch in the $yz$ plane.
(6) Draw another control point spline, with distance 100 in the $y$ direction and variance between 0 and 1 meter in the $z$ direction.
(7) Keeping the old copy, translate in the $y$ direction by 100 meters.
(8) Change the environment from "Model" to "Patch."
(9) Extrude either the entire second sketch, say, 50 meters in $x$ direction (symmetrically), or each line individually and stitch the resulting curved surfaces (does not matter which option).
(10) Translate this curved surface 1 meter in the $z$ direction.
(11) Move back to the "Model" environment.
(12) Under "Modify", choose "Split Body," where the body to split is the first body, and the splitting tool is the second body (the curved surface).
(13) Hide all resulting bodies, except the first (which is the top portion of the original first body).

As stated, our 3D domain yields our 2D domain by $xy$ projection, which is 200 meters in length and 30 meters in width. Our second domain, with obstacles, was obtained in COMSOL. Here, we excised 24 circles with a radius of 1 meter by placing them on our domain in the desired spots and taking the symmetric difference with the original domain. Figure 4 depicts our domains, as imported into COMSOL. For the remainder of this section, the problem with the domain without obstacle will be called "Problem A," and the problem for the domain with obstacles will be called "Problem B." Note that the domain with the prescribed dimensions, a set temperature of 68° F, and other standard properties of water will induce a turbulent regime, with Re≈6e7 at the mouth of the river when the inflow ramps up to its maximum of 2 m/s.

2.2. **COMSOL Implementation.** In COMSOL, we attempted to use the turbulence module in 3D and utilized both the turbulence and particle tracing modules in 2D. We will discuss only the 2D set-up, as the turbulence set-up is analogous for 3D. The process is outlined in the flow-chart in Figure 5, and it will make references to edges in figure 4.

In 2D, we first solved for the flow with the turbulence model, then used this solution for the particle tracing. We will provide the general steps for solving Problem A, as Problem B is entirely analogous, with obvious boundary conditions on the boundaries of the disks. To summarize the set-up, we imported our domain and set the material of the domain to be water. For the turbulence, we set the water to be at rest. We chose an inlet on the far left edge from Figure 4(a) and utilized a ramp function with a slope of 2, to avoid discontinuities in the flow. The outlet was on the far right, and it had a zero pressure condition, with backflow suppressed (default). All other boundaries had no-slip wall conditions. For the particle tracing, we chose solid particles with a density of 2200 kg/m$^3$, with a diameter of 1 $\mu$m (defaults). We used standard drag correlations for the drag force, with the velocity given by the velocity field of the flow. We had an inlet condition on the same edge as previous, where 15 particles were released every 10 seconds for 150 seconds, with initial velocity given by the flow. An outlet condition on the same edge as before was also used, where the particles would disappear. All other edges had a wall condition, where particles had a 50% chance to bounce and a 50% chance to stick. Both problems were solved for 150 seconds, with output every second. As stated previously, the turbulence model was solved first, then the velocity field was used for the particle tracing. After completing these simulations, we performed our post-processing in COMSOL, as well. This was comprised of five features, velocity magnitude over the domain with both a high and low upper bound, vorticity magnitude, streamlines, and particle trajectories. These will be discussed more in-depth in the subsequent sections.

## 3. Results and Discussion

3.1. **3D Failures.** In 3D, we attempted to implement the steps of the previous section to the geometry in Figure 3 without coupled particle tracing, which we planned to perform in post-processing. However, this problem did not prove to be tractable. Despite utilizing Longleaf to run simulations on their big data nodes, results would fail either in the stationary portion or very early in the time-dependent solver, or I would even not reach these stages in 5 days of runtime. In this process, we varied the mesh from normal to extra fine and the solver from iterative to direct. There are a few likely culprits in this failure:

(1) **Domain:** While the domain was not very complicated, it was extremely large (horizontally), and 3D. This made the problem to solve very large. The large discrepancy in vertical versus horizontal scaling led to a requirement of a very fine mesh in order to hope for convergence, which added to the computational intensity.
(2) **Turbulence:** Turbulence is a very difficult and expensive computation in general. The size of the domain led to a very large Reynolds number, causing our regime to be highly turbulent. The thinness of our domain can mathematically lead to chaotic behavior, which can physically manifest as added turbulence.

For these reasons, we switched to a 2D model.

3.2. **2D Successes.** As stated in the COMSOL implementation, we examined five quantities to discern key characteristics of the flow and attempt to solve our problem in some meaningful way. All of these will be provided as plots which are superimposed on the geometry at a given time slice. Figures 6-7 (resp. 8-9) are plots of the velocity magnitude, with a moderate (resp. low) upper bound. The moderate upper bound gives a more well-rounded overview of the velocity of the fluid, since higher upper bounds tend to blend too much of the flow together. Low velocity is useful to demonstrate the dispersion of the fluid, which starts at rest. The next quantity extracted was the vorticity magnitude, which also helps capture dispersion and general behavior of the flow. For a similar reason, we plotted streamlines as our fourth component. Finally, we will provide the particle tracing results. As stated earlier, simulations were run for 150 seconds. We will begin by describing the case of no obstacles, then move on to the obstacle case, where we will contrast with the former.

3.2.1. *No Obstacles.* The first portion of results are related to velocity with a moderate upper bound. Here, the magnitude ranged from 0 m/s to an 8 m/s (that is, the upper bound on the color bar is 8), although the actual flow is contained in $[0, 10.37881]$ m/s, with minimum 0.01568 m/s after 150 seconds. The relevant figures are contained in Figures 6-7. It is worth noting that the peak velocity from the inlet is 2 m/s, meaning that the velocity increases by, time, more than a factor of 5. Due to the winding nature, the velocity reaches its peak adjacent to the wall shortly after the strong turn in the middle of the figure. Note that by 133 seconds, the flow has settled into a fixed profile. We are most interested in behavior in this regime, as this is when we see the true profile of the river for an inlet with a constant inflow of 2 m/s. Various subfigures feature re-circulation, due to portions of the fluid striking the boundaries (see Time=16, 33, and 50 seconds).

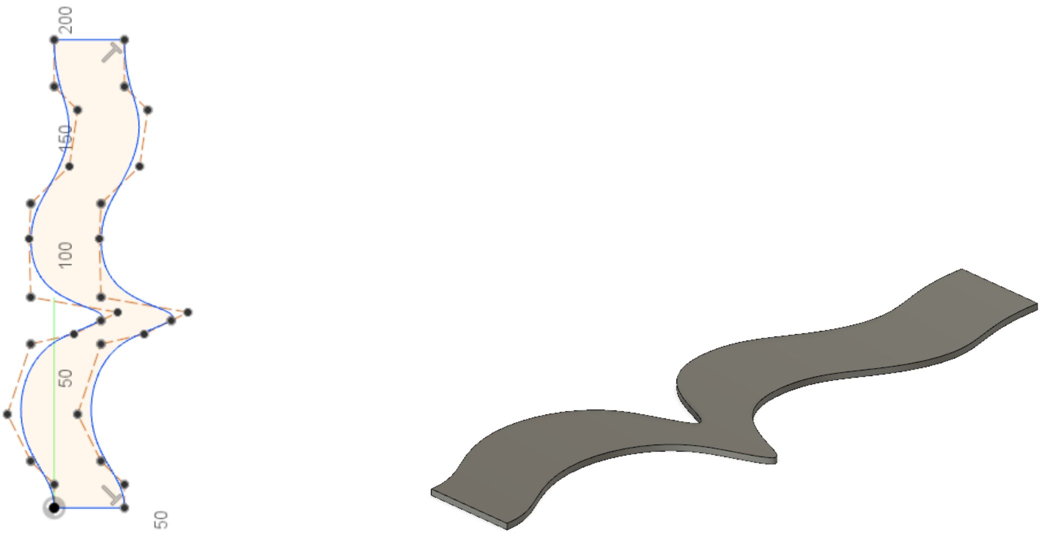
Due to the high upper bound on this figure, it is not clear how well-dispersed the flow from the inlet becomes. For the velocity plots with a low velocity upper bound, we placed the maximum at 0.2 m/s. The accompanying figures are Figures 8-9. These intimate that the fluid has dispersed through the entire domain, which is not apparent from the earlier figures. For example Figures 6-7 does not clearly show that the velocity has non-zero magnitude in the right corner at $y = 80$ m, whereas Figures 8-9 indicate that, except for a near-singular neighborhood in the middle, the velocity is 0.2 m/s or more in magnitude (by singular neighborhood, we mean a neighborhood of a singular point, and analogous for near-singular). Almost certainly, this neighborhood contains a single singular point, but this is not picked up due to the discreteness of the domain (i.e. due to the lack of continuity in computing). Hence, we will call this neighborhood a singular neighborhood and the point a singularity. The only other singular neighborhood of the flow is the strip from approximately $y = 110$ to $y = 190$, with $x$ between 0 and 20 (again, approximately). Once again, it is likely that these all represent fixed points of the flow. We will refer to the entire strip as singular, although it appears to be comprised of discrete set of singularities (here, when we say discrete, we mean finite). Figures 12-13 help explain this phenomena. Observing the streamlines in the last two frames of Figure 13, we can see that, first, the flow has indeed settled into a set profile. In particular, we can see that there is rotation about these singularities, although the precise type is unclear, as is the stability. We can also see the evolution of fluid lobes exhibiting rotational behavior in this figure, although figure. Figures 10-11 demonstrate this much more clearly. For example, we can see shortly after the fluid strikes the winding portion of the boundary in the middle, a vortex breaks off and propels upwards. Then, it strikes the wall a second time and re-circles downward. The profile of vorticity of the flow settles similar to that of the velocity. This behavior is seen to affect the particles in Figures 14-15, where they re-circulate throughout the domain after hitting these areas. They follow the flow fairly orderly, with bouncing having a fairly minimal effect, with one exception; striking the boundary towards the top left of the domain often leads to re-circulation, as particles enter the area near the singular strip. Since the particles follow the streamlines (unless there is boundary collision) and all particles enter from the inlet, no particles enter into the corner near $(40, 82)$ (no collision will allow a particle to enter here, based on the streamline behavior), and this will not change will the addition of more particles. Hence, particle trajectories are non-dense in the domain. Finally, note that once the flow has settled, there re-circulation still occurs.

3.2.2. *Obstacles.* Let us call the case of no obstacles "NO," for convenience. All of the corresponding figures are directly below those for the NO case (Figures 16-25). In the case of the introduction of obstacles, the magnitude of the velocity reached as high as 11.68275 m/s, faster than in the NO case, despite the same initial values and inlet condition. The peak of the magnitude occurs around obstacles in the same region as the other case. While the obstacles in the top right of the domain prevent the flow profile from settling fully in that region, the general profile settles much faster than in the NO case. Furthermore, it settles into a very similar profile, with the main differences being small fluctuations caused by obstacles in path of the primary current. The singular strip featured previously is disconnected by the obstacles in that region, leading to increasingly disjoint singular neighborhoods about many of the same fixed points. The main vortex shown previously in the vorticity plots is fractured earlier before it makes contact with the wall boundary here, due to the obstacles. Further, each obstacle induces an area of high vorticity in its wake. The peak vorticity is much higher in this case; the maximum vorticity magnitude here is approximately 64.67013 1/s, while it is approximately 34.53654 1/s in the NO case. In the O case, this occurs on the boundary of an obstacle around the point (15,110), whereas in occurs on bank above the sharpest wind in the river in the NO case, near the point (20,85). Despite this, the general values of the vorticity are consistent, and the vorticity still "settles" in analogous to that of the NO case, just as with their speeds. The streamlines corroborate the conclusions drawn from the velocity and vorticity plots. Since the vortex was broken up earlier, this leads less time for particles have a chance to re-circulate, which is supported by the particle trajectories. Far fewer particles re-circulate, but like previously, this does not completely stop once the flow profile becomes more settled. Additionally, particles still get captured by the singular neighborhoods, leading to a similar amount of deposition.

3.3. **Summary of 2D Comparisons and Conclusions.** Let us summarize the comparisons made in the previous sections between the no obstacle ("NO") and obstacle ("O") cases. In studying their velocities, we saw that both profiles generally settled, with the NO case being more settled and the O case, generally, settling faster. The speeds were similar, with the speed reaching higher maximum values in the O case, which occurred as the fluid passed obstacles. The inlet flow dispersed well throughout the domain, with the exception of a discrete set of fixed points, in each case. In the O case, the velocity surrounding the singularities in the top left of the domain increased to the mean more quickly. Their vorticity profiles were analogous in the same sense as the velocity profiles. Once again, the O case had a higher maximum magnitude, which occurred on the boundary of the obstacles. Their streamlines told a similar story, although they were "tighter" in the O case, due to the need to avoid collision with the obstacles. Finally, in the case of particle tracing, the obstacles did not have a significant effect in terms of aiding in the dispersion of the particles. More re-circulation actually occurred in the NO case, as the obstacles accelerated the breaking of a large vortex, preventing particles from being manipulated by it, unlike in the NO case. In both cases re-circulation does not completely stop once the flow has settled. Furthermore, a similar amount of particles got caught by the singular neighborhoods of the flow in the top right of the geometry, meaning that deposition was comparable. These observations support the conclusion that, in our simulations, the obstacles did not have a significant impact on the dispersion or deposition of particles.

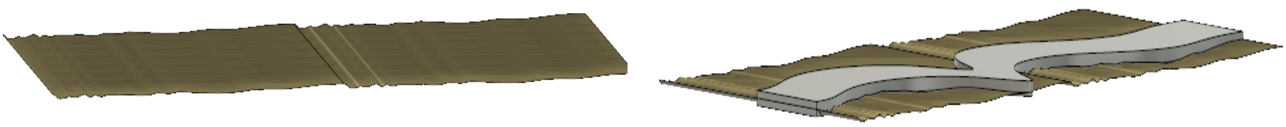
3.4. **Future Work.** There are numerous avenues of future work. The most obvious, given the initial goal of the problem, it is extend to 3D simulations. After seeing the steps required to work on a 2D model, an extension to the harder case seems more attainable. It may be more effective to use something other than COMSOL, though, as it is likely not optimized to handle such a problem. Another idea is to actively track erosion. COMSOL possesses this feature in their particle tracing module, but I did not get a chance to learn how to use it properly. This would yield a much more realistic representation of the scenario. For added realism, one could use actual data to model the river (the dimensions, the velocity, the temperature, etc.). One last concept would be to model forming a river into an estuary. Since estuaries have higher salinity, it would be interesting to see how this affects the flow.

## 4. Figures



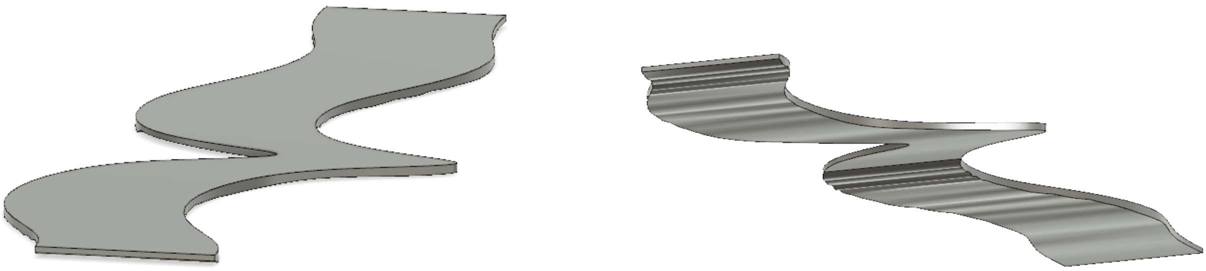

(a) (b)

Figure 1. Figures for steps 1-4: (a) control-point-spline-based $xy$ sketch, (b) Extruded $xy$ sketch



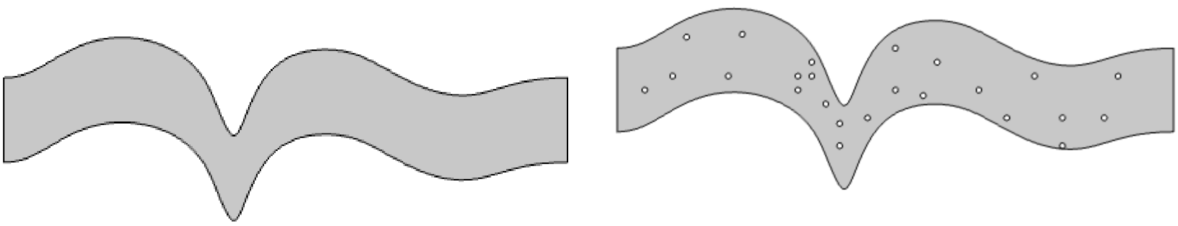
(a) (b)

Figure 2. Steps 5-10: (a) Curved surface resulting from control-point-spline-based $yz$ sketch, (b) Intersection of the two bodies after translating the curved surface



(a) (b)

FIGURE 3. Steps 11-13: (a) Overhead view of desired domain after splitting is performed, (b) Underneath view of desired domain after splitting is performed



(a) (b)

FIGURE 4. 2D domains: (a) No obstacles, (b) Obstacles

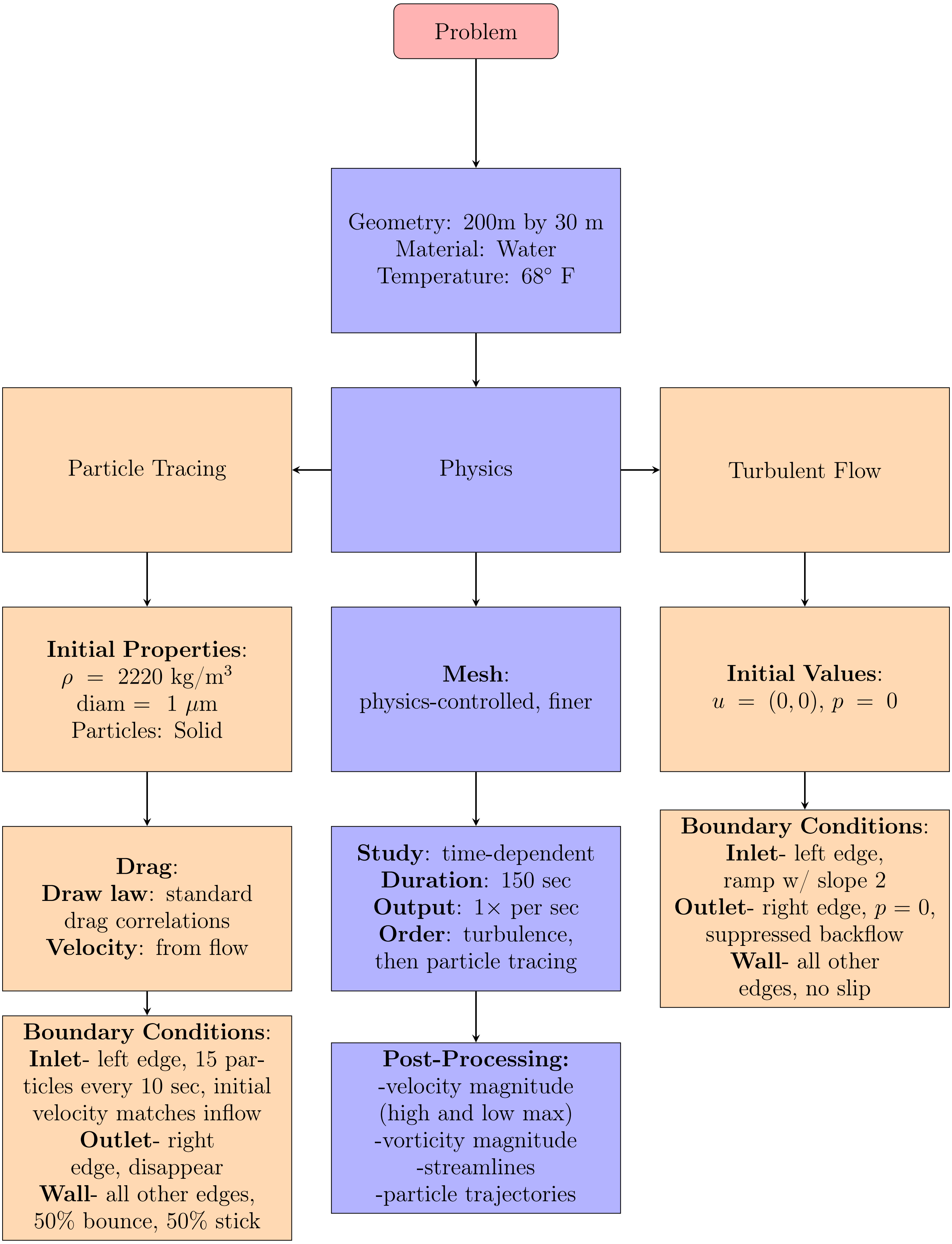



FIGURE 5. Flow-chart of 2D COMSOL Implementation

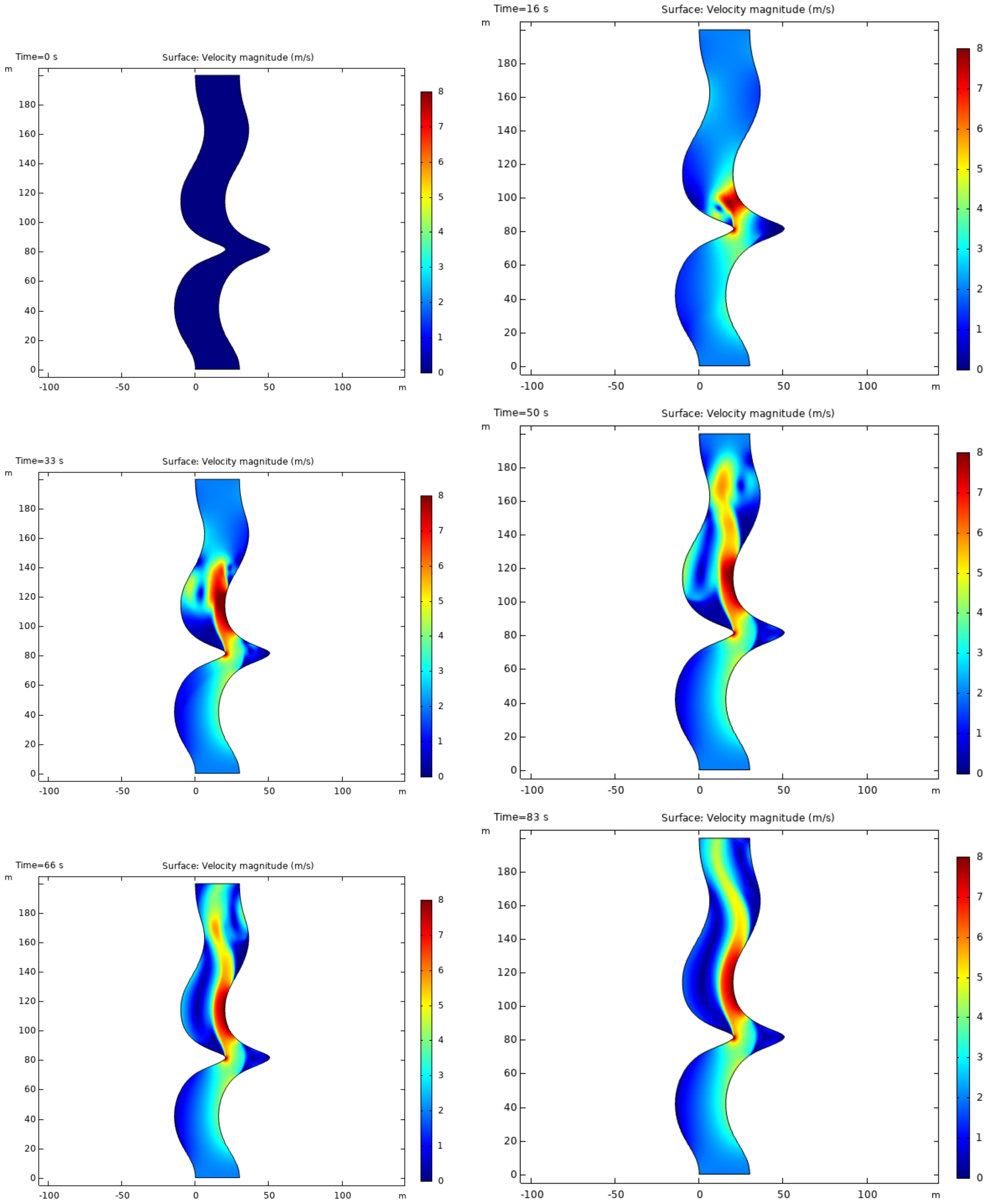



FIGURE 6. Velocity magnitude evolution over 150 s, no obstacles (moderate upper bound of 8 m/s)

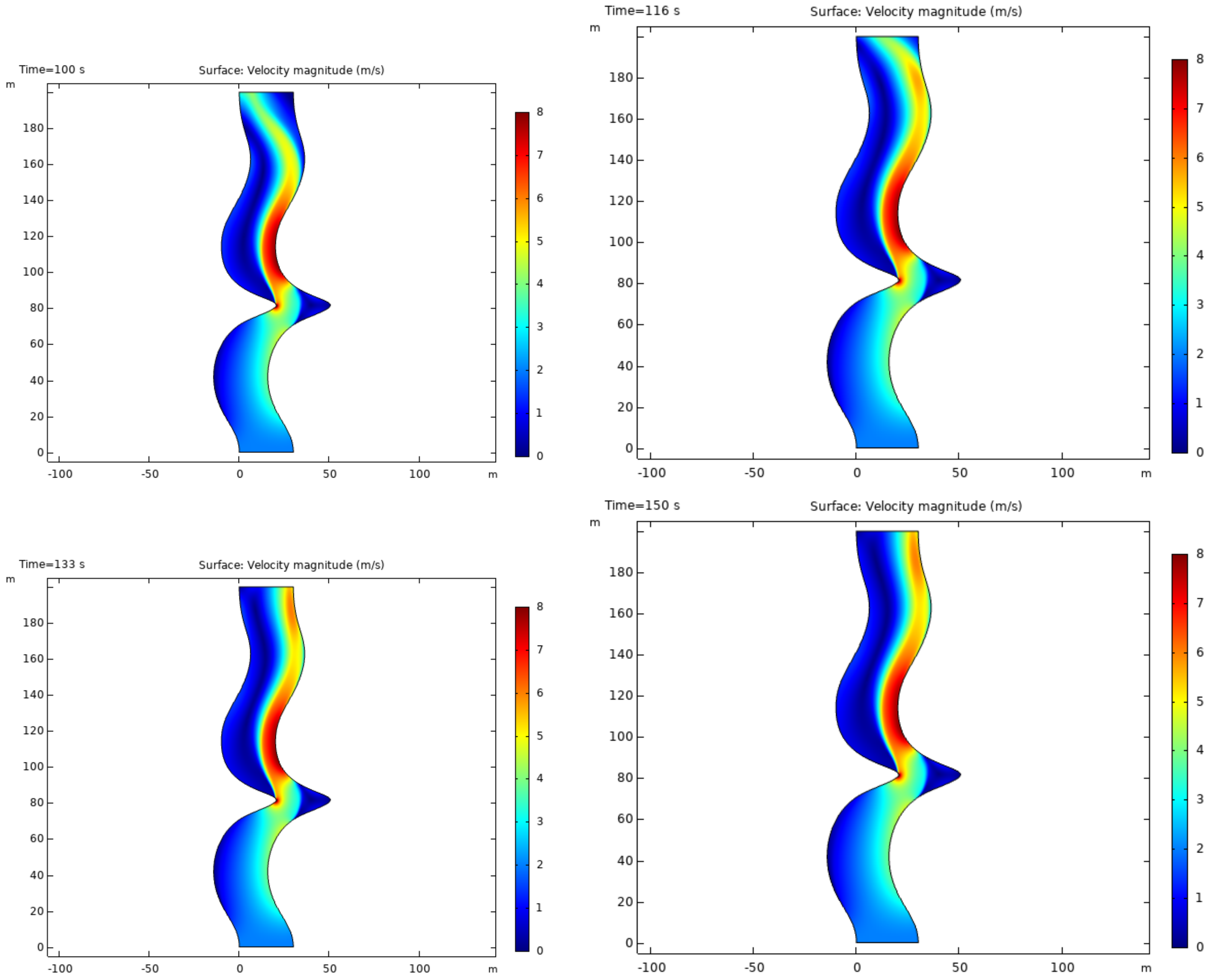



FIGURE 7. Velocity magnitude evolution over 150 s, no obstacles (moderate upper bound of 8 m/s)

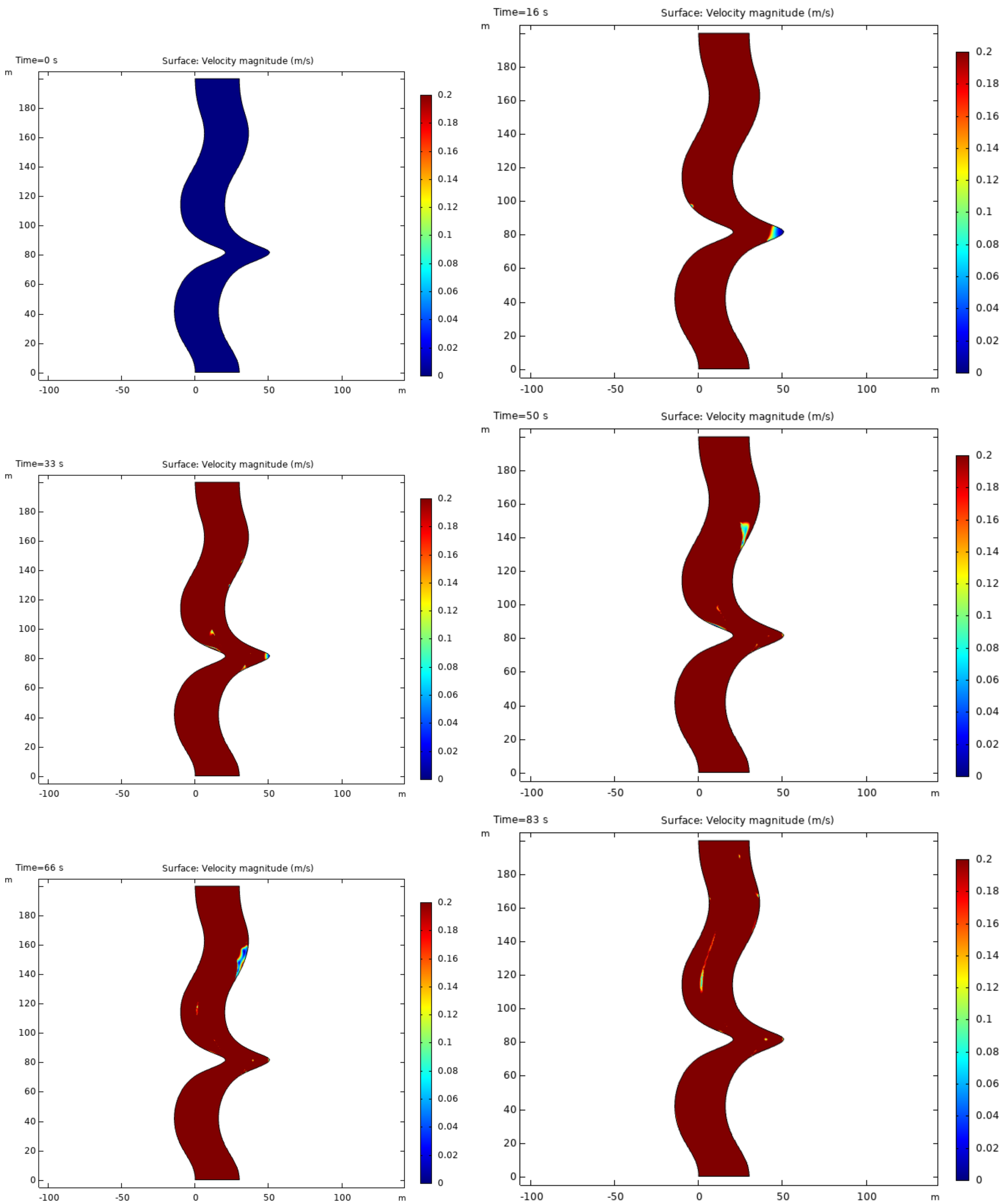



FIGURE 8. Velocity magnitude evolution over 150 s, no obstacles (low upper bound of 0.2 m/s)

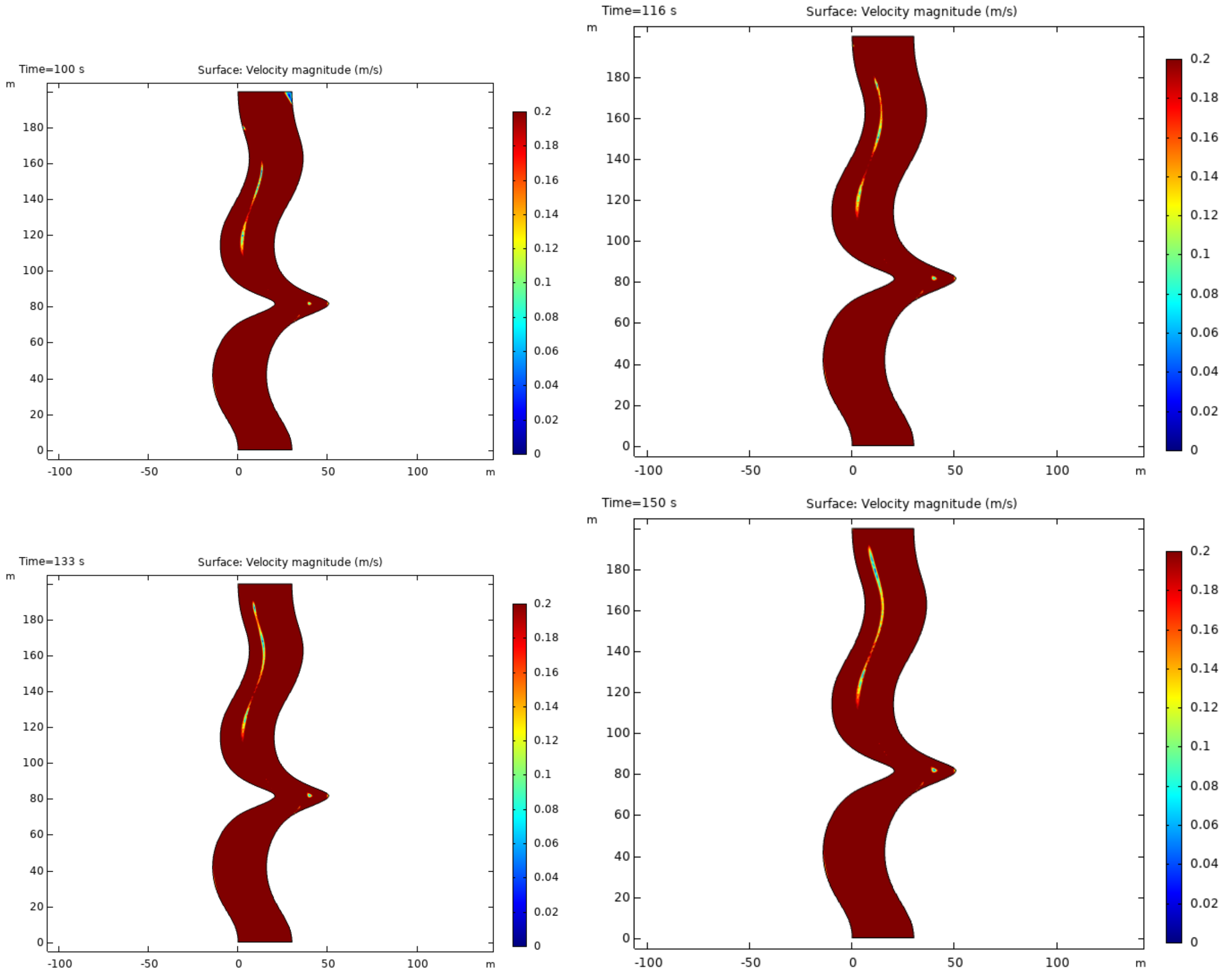



FIGURE 9. Velocity magnitude evolution over 150 s, no obstacles (low upper bound of 0.2 m/s)

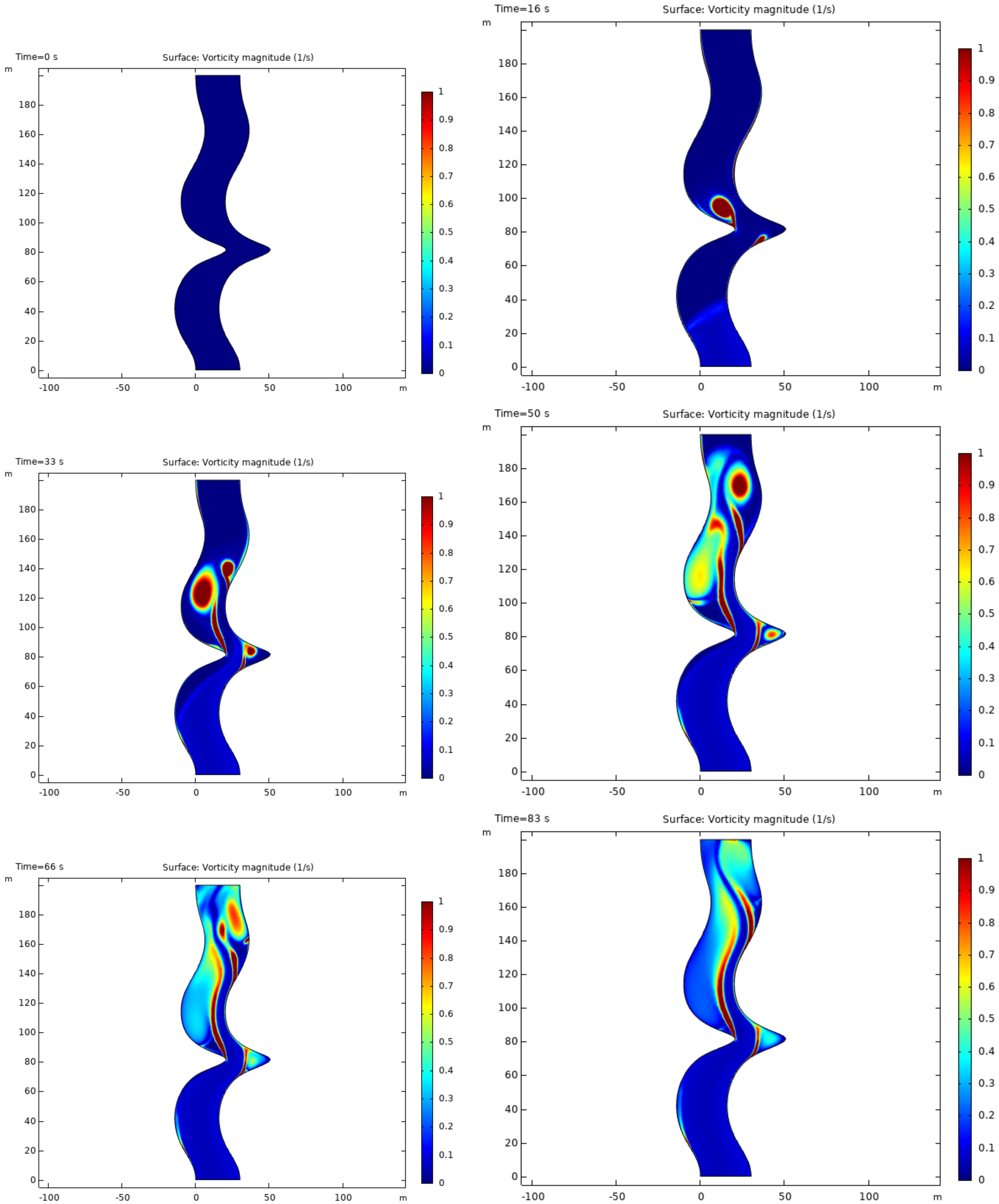



FIGURE 10. Vorticity magnitude evolution over 150 s, no obstacles

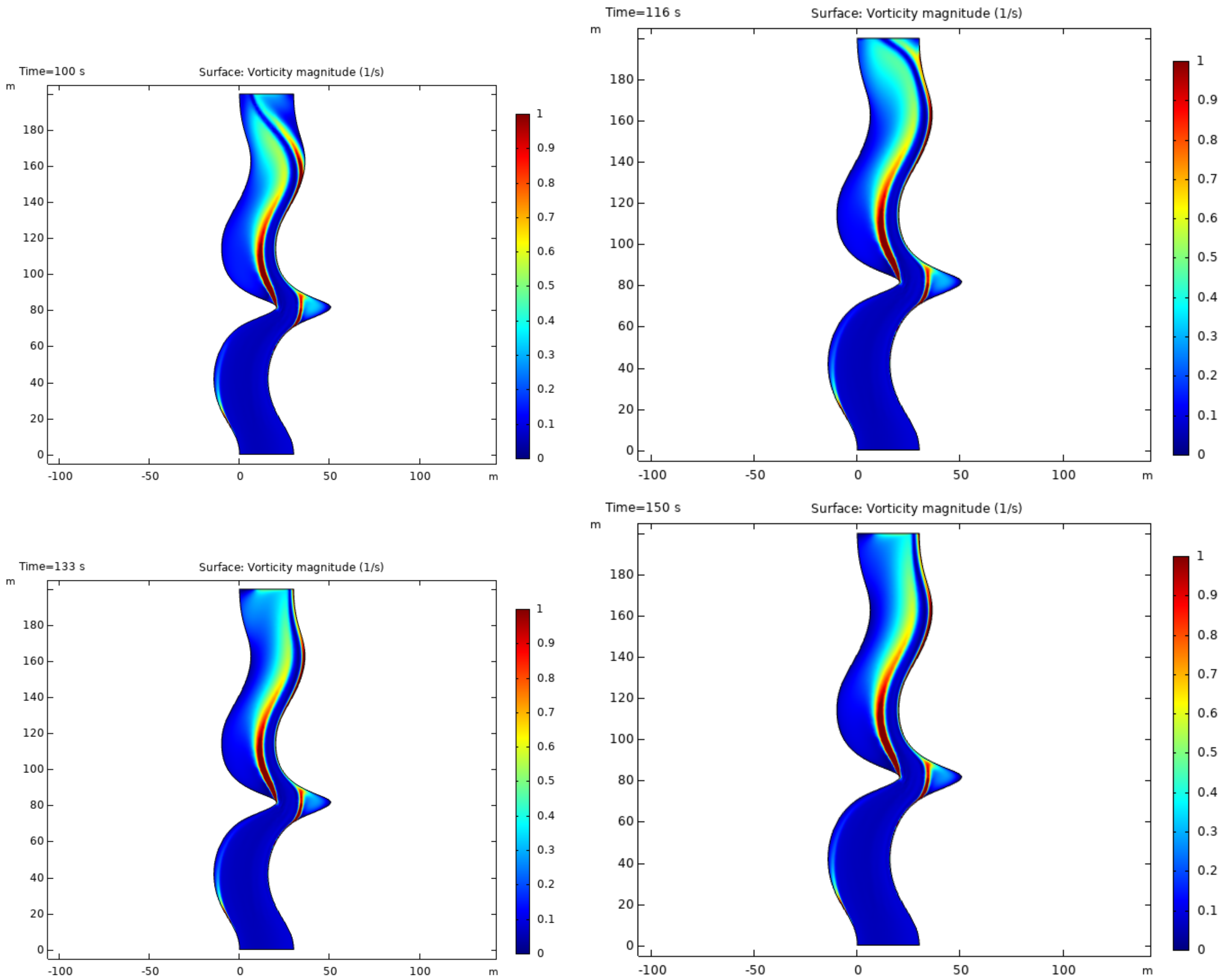



FIGURE 11. Vorticity magnitude evolution over 150 s, no obstacles

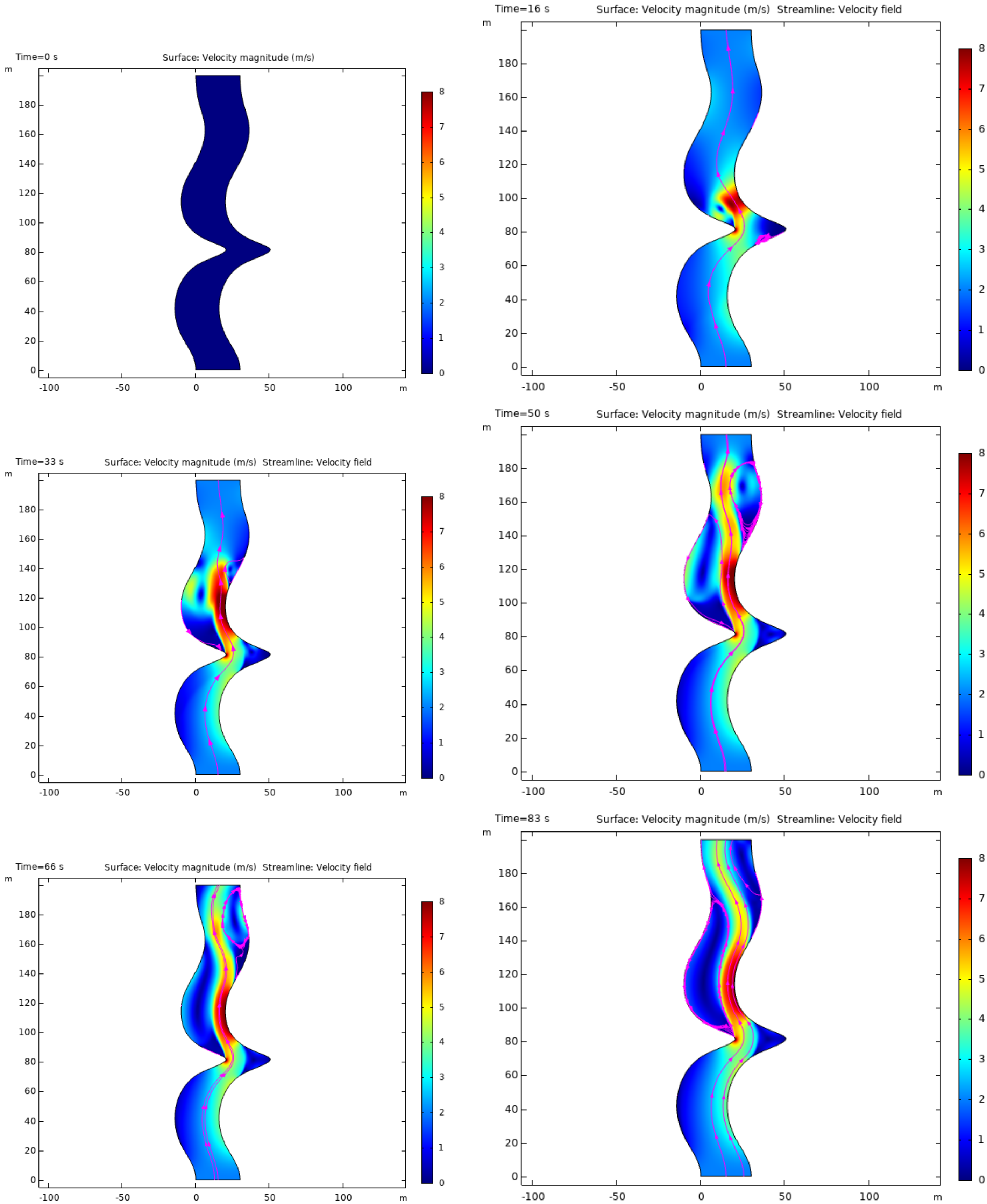



FIGURE 12. 35 streamlines superimposed over velocity magnitude evolution over 150 s, no obstacles

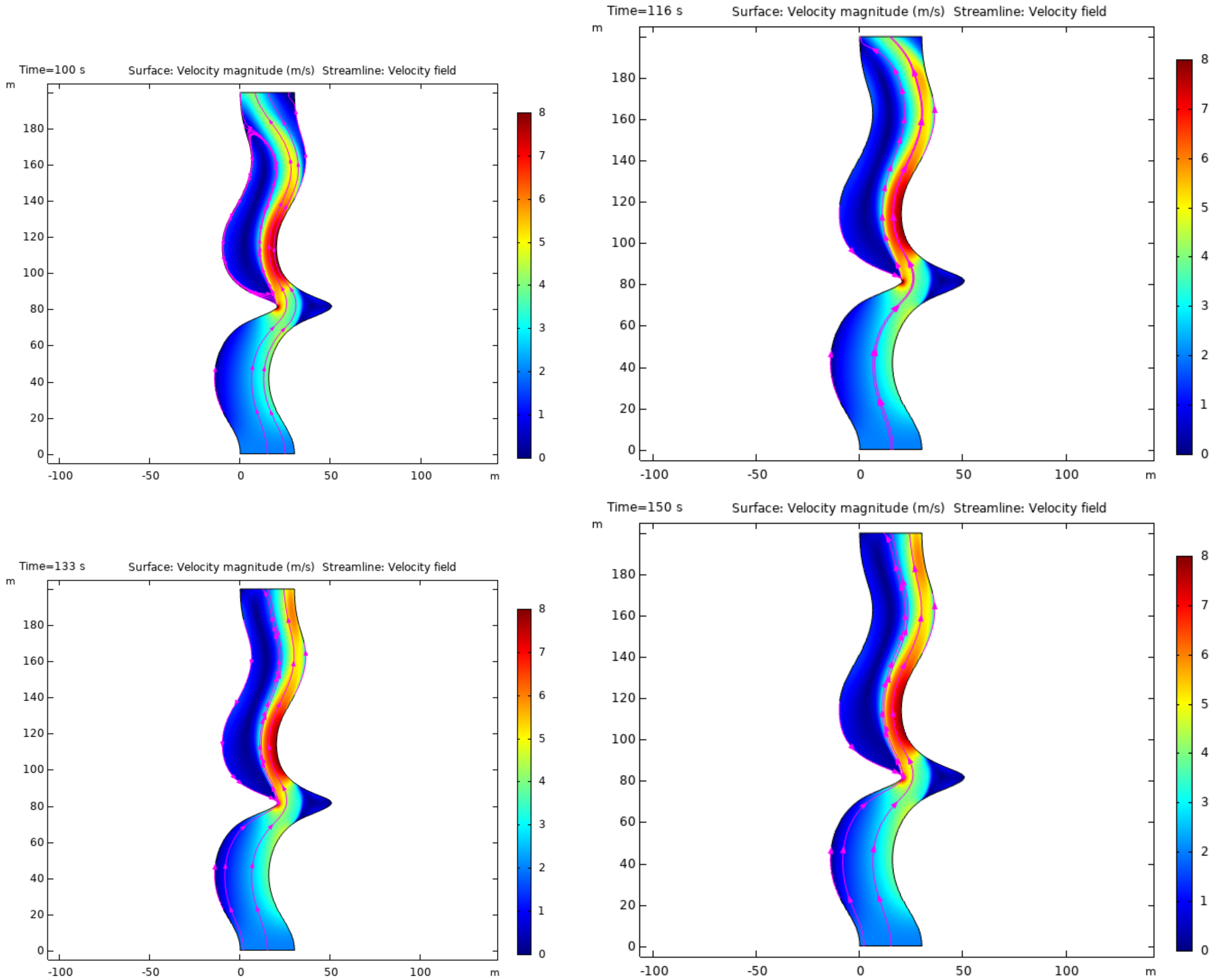



FIGURE 13. 35 streamlines superimposed over velocity magnitude evolution over 150 s, no obstacles

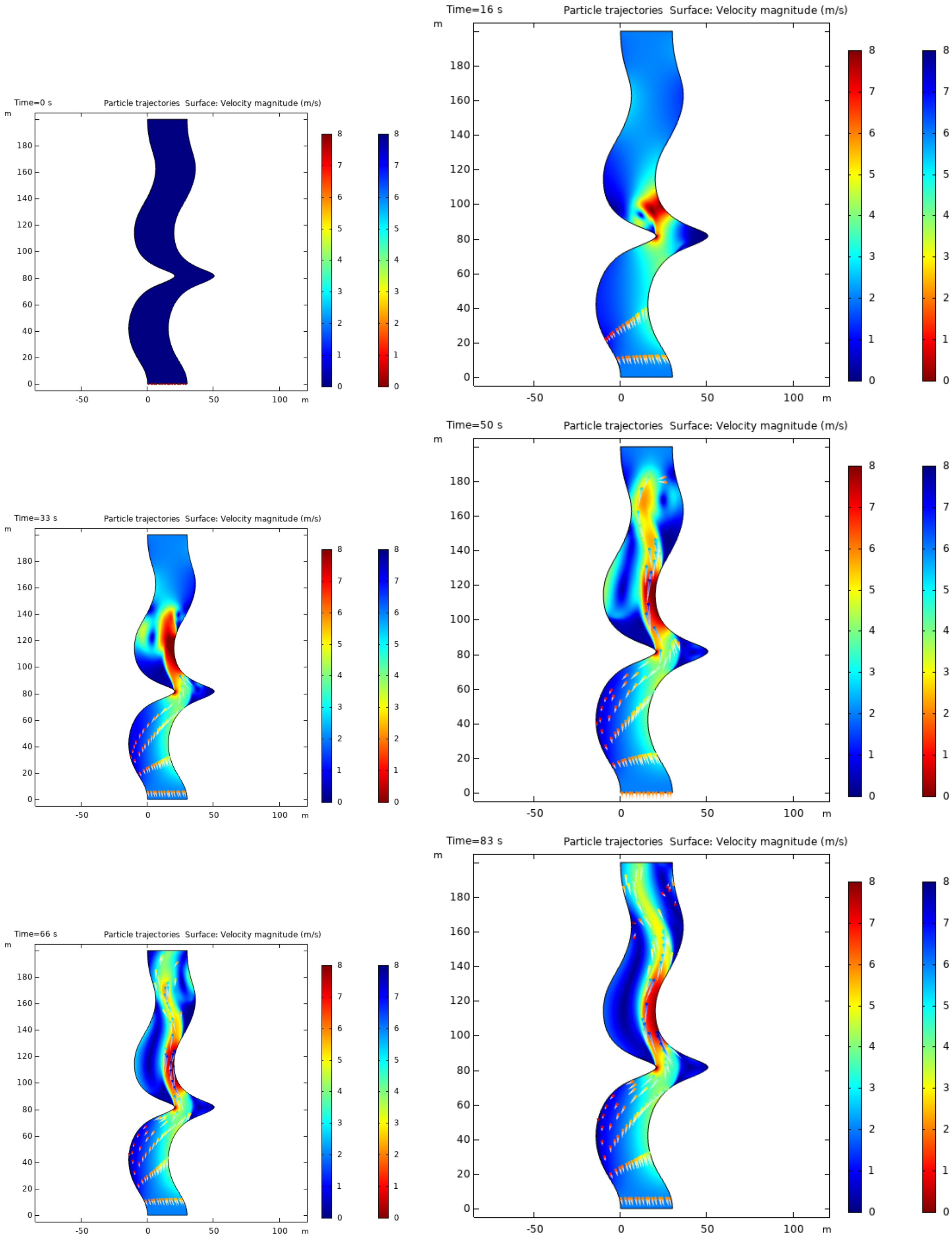



FIGURE 14. Particle trajectories superimposed over velocity magnitude evolution over 150 s, no obstacles

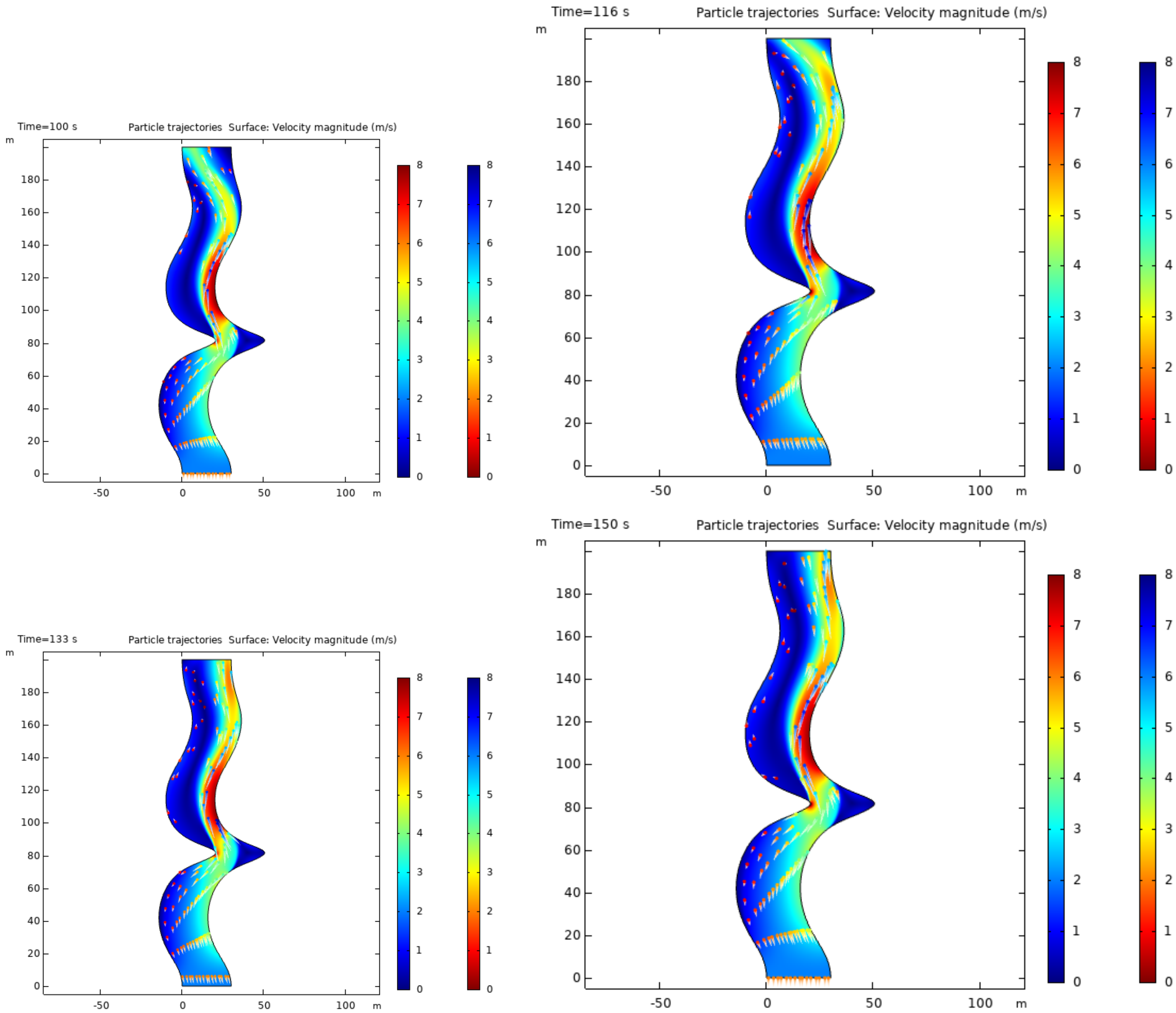



FIGURE 15. Particle trajectories superimposed over velocity magnitude evolution over 150 s, no obstacles

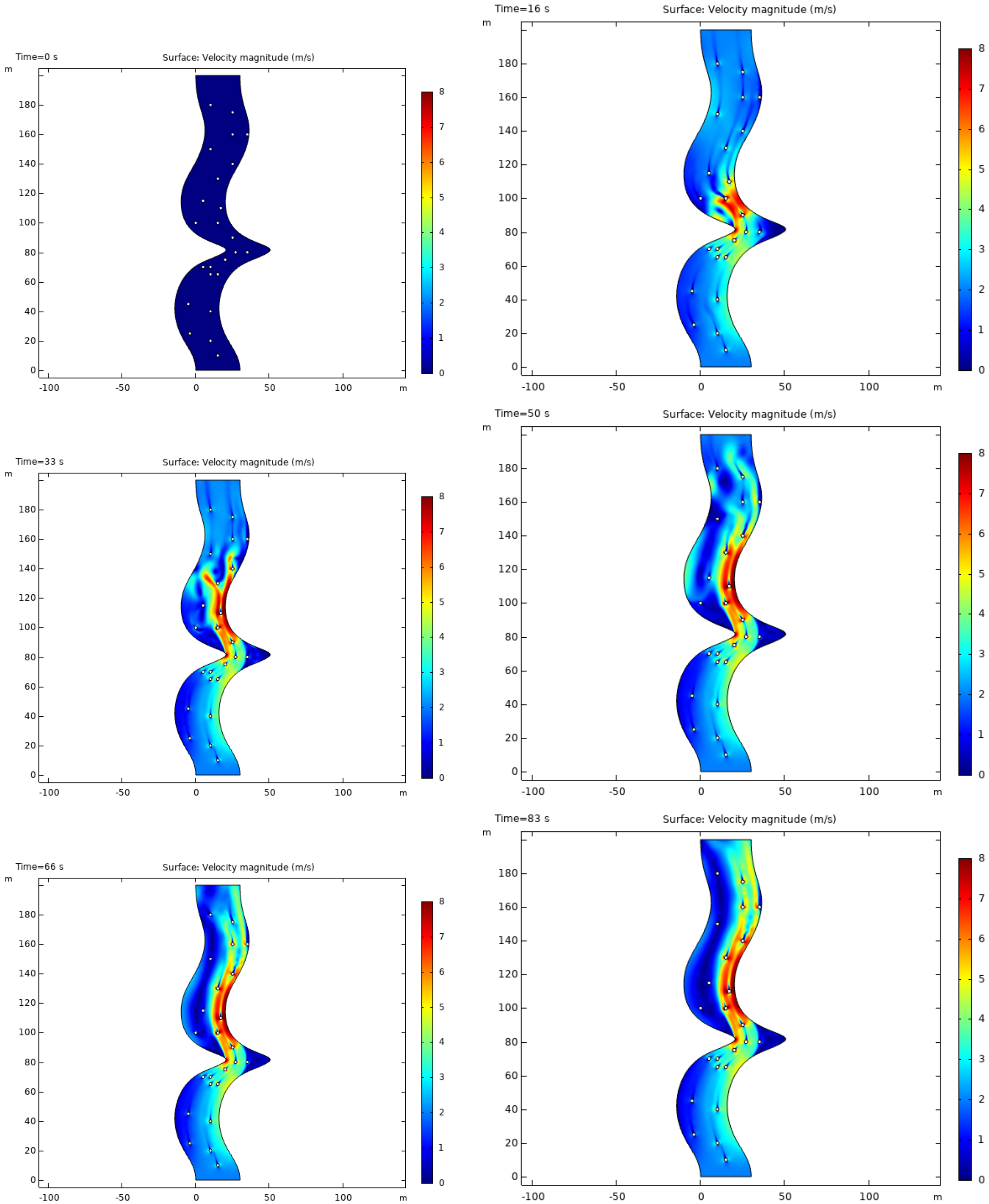



FIGURE 16. Velocity magnitude evolution over 150 s, obstacles (moderate upper bound of 8 m/s)

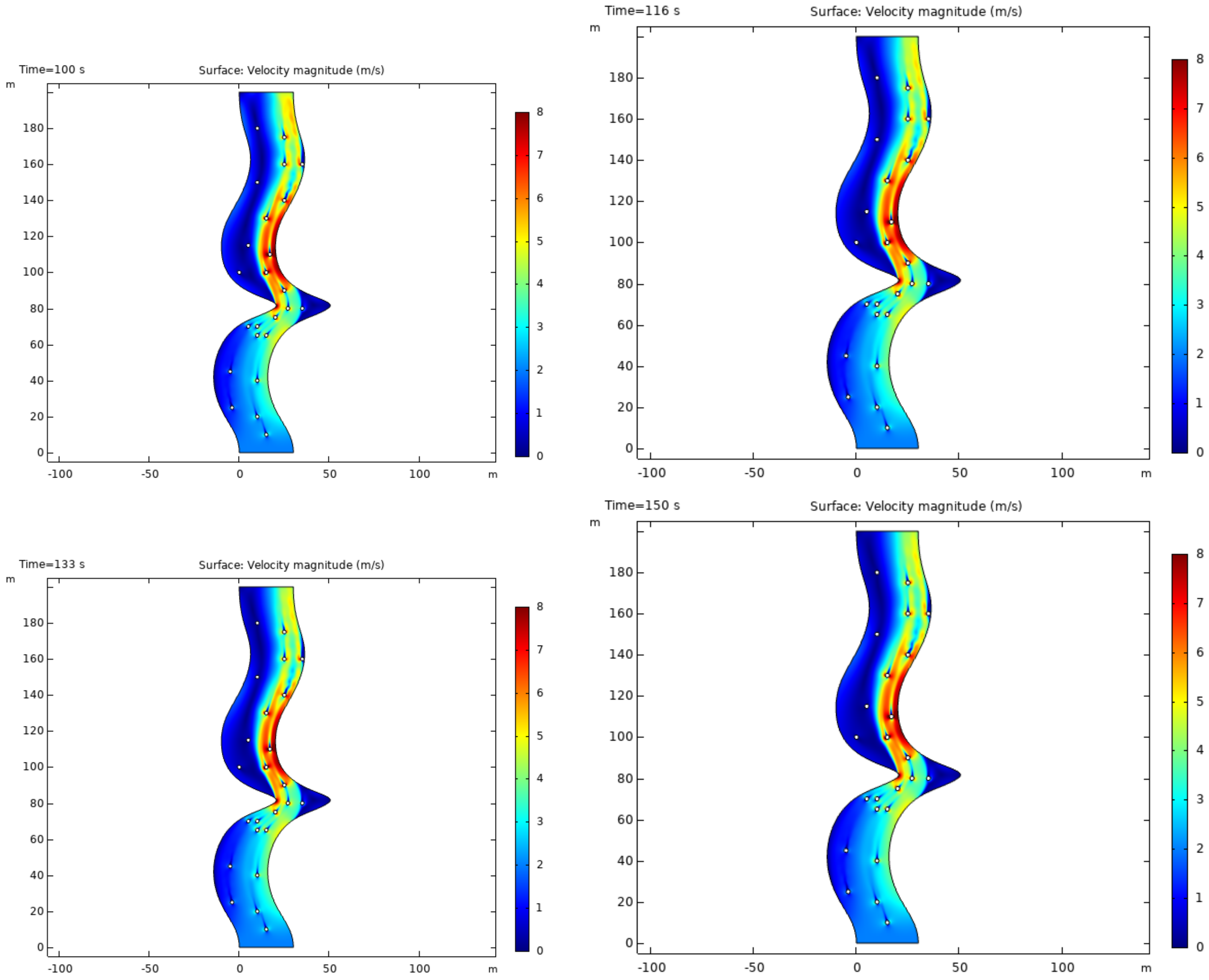



FIGURE 17. Velocity magnitude evolution over 150 s, obstacles (moderate upper bound of 8 m/s)

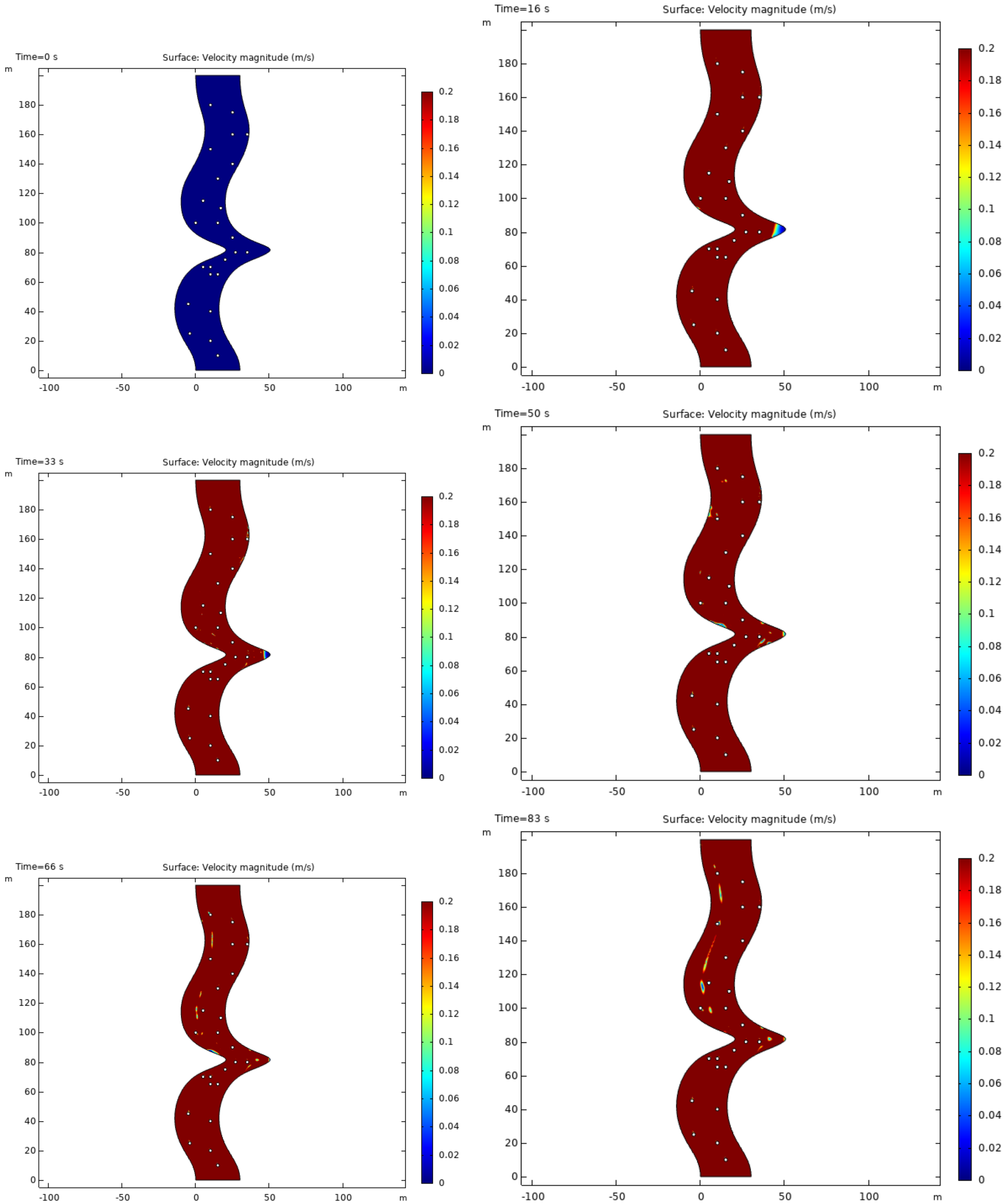



FIGURE 18. Velocity magnitude evolution over 150 s, obstacles (low upper bound of 0.2 m/s)

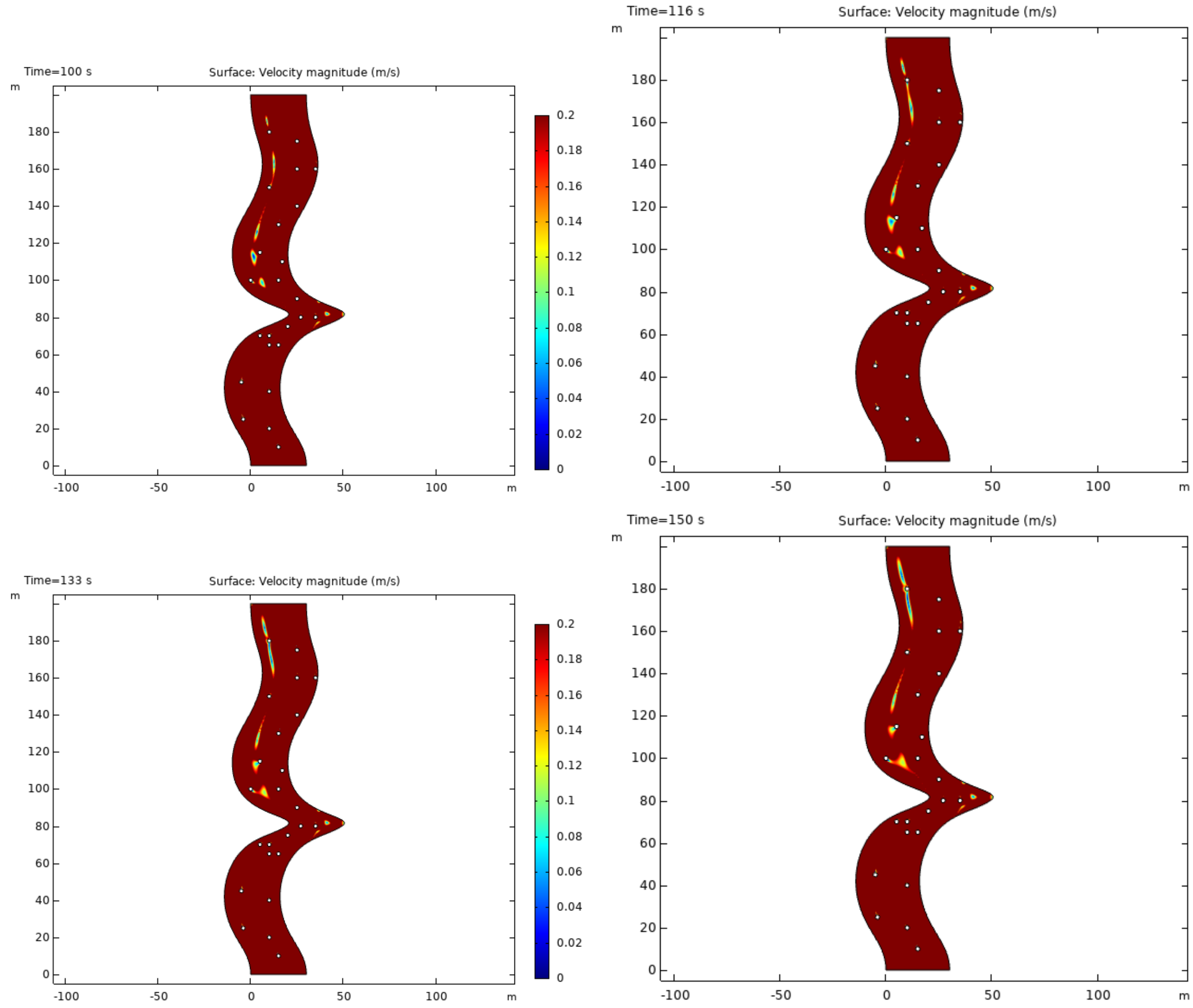



FIGURE 19. Velocity magnitude evolution over 150 s, obstacles (low upper bound of 0.2 m/s)

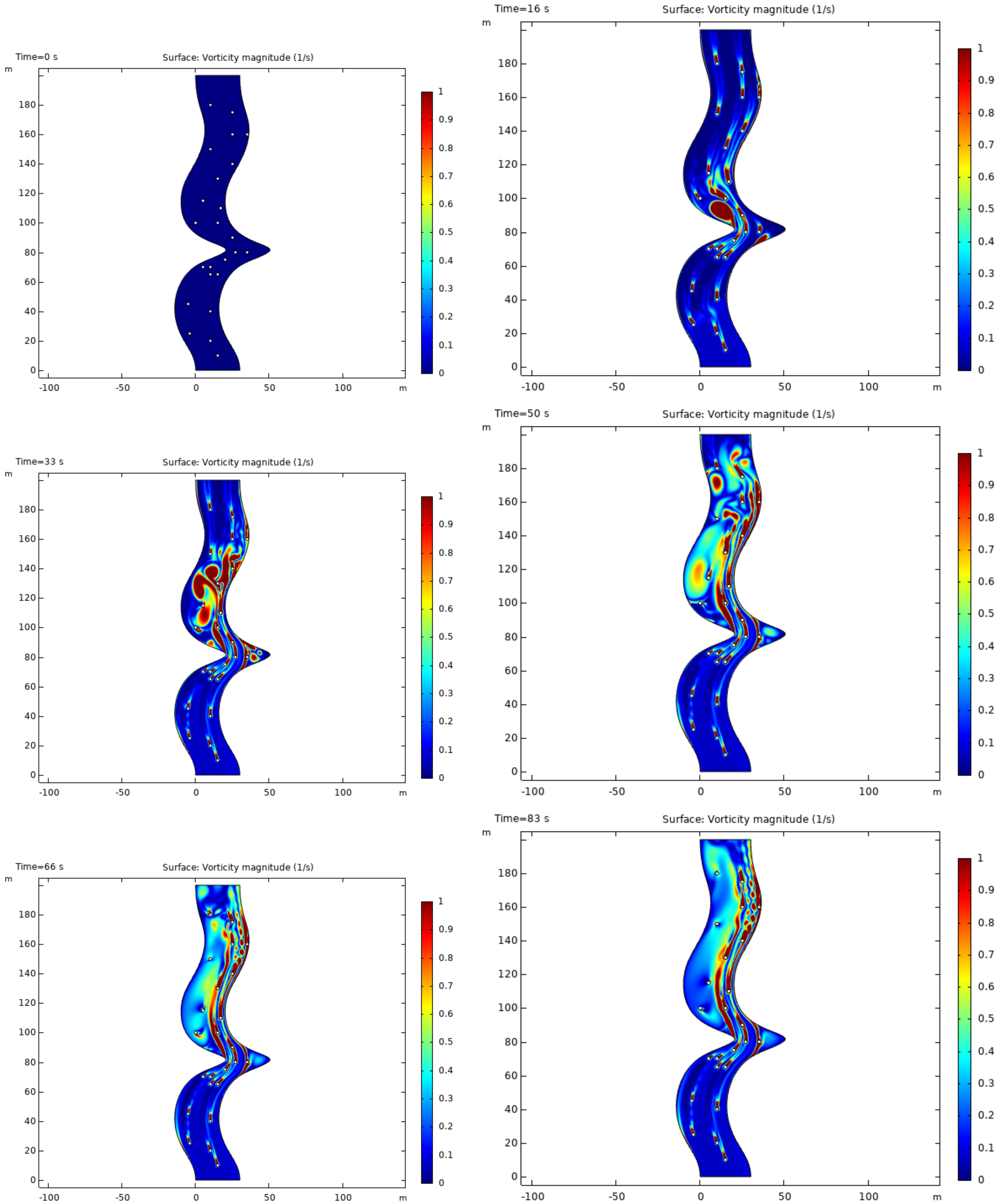



FIGURE 20. Vorticity magnitude evolution over 150 s, obstacles

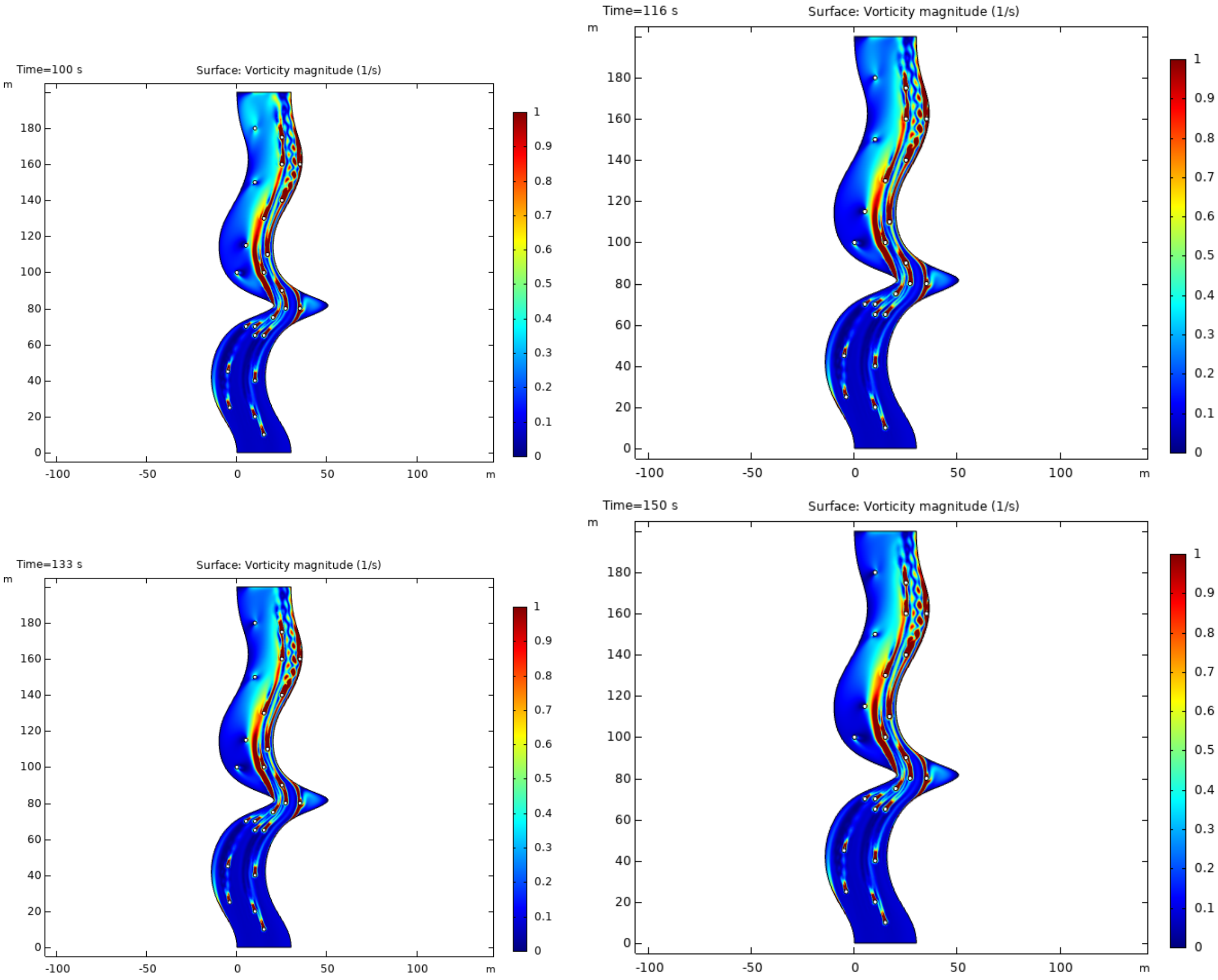



FIGURE 21. Vorticity magnitude evolution over 150 s, obstacles

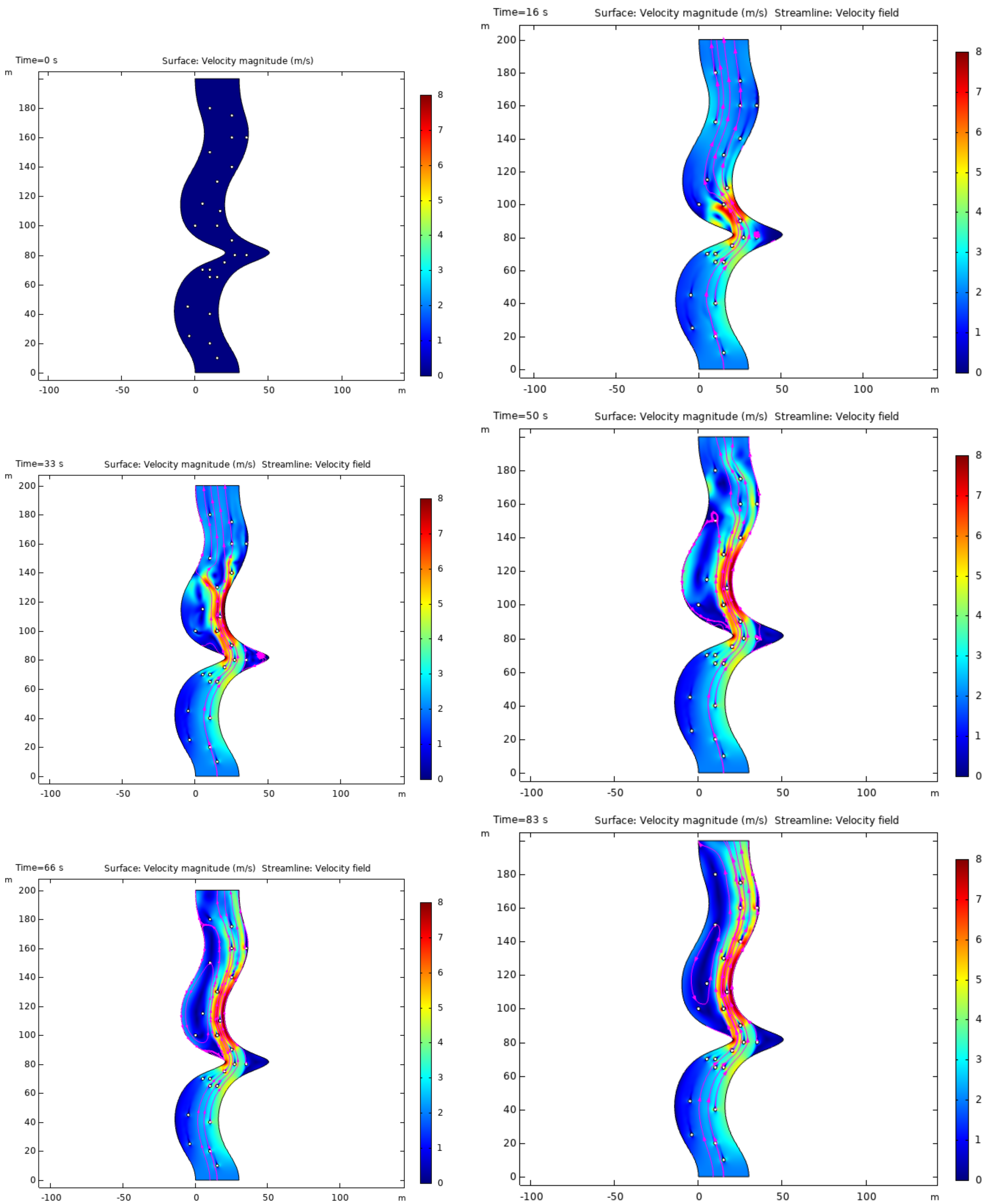



FIGURE 22. 35 streamlines superimposed over velocity magnitude evolution over 150 s, obstacles

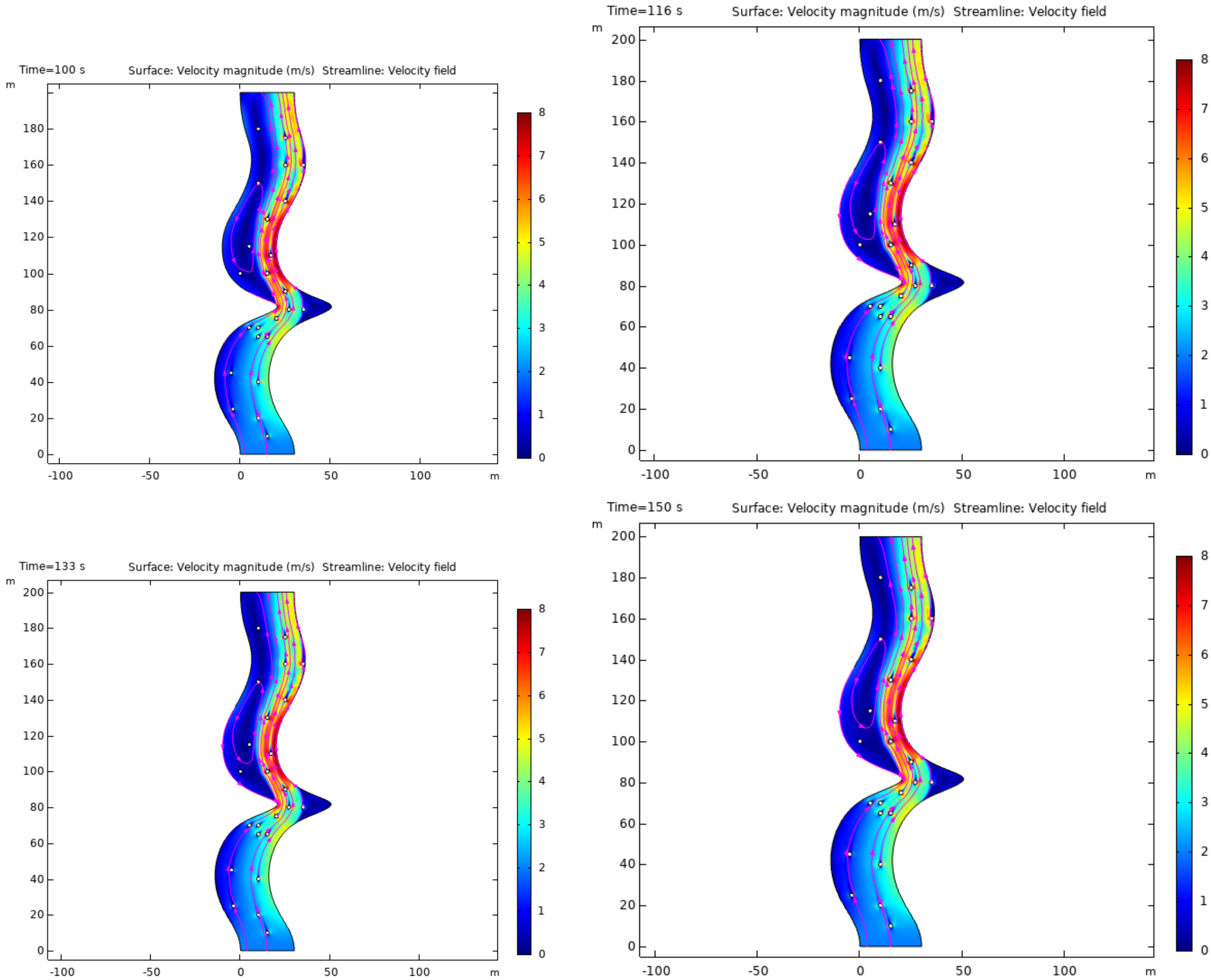



FIGURE 23. 35 streamlines superimposed over velocity magnitude evolution over 150 s, obstacles

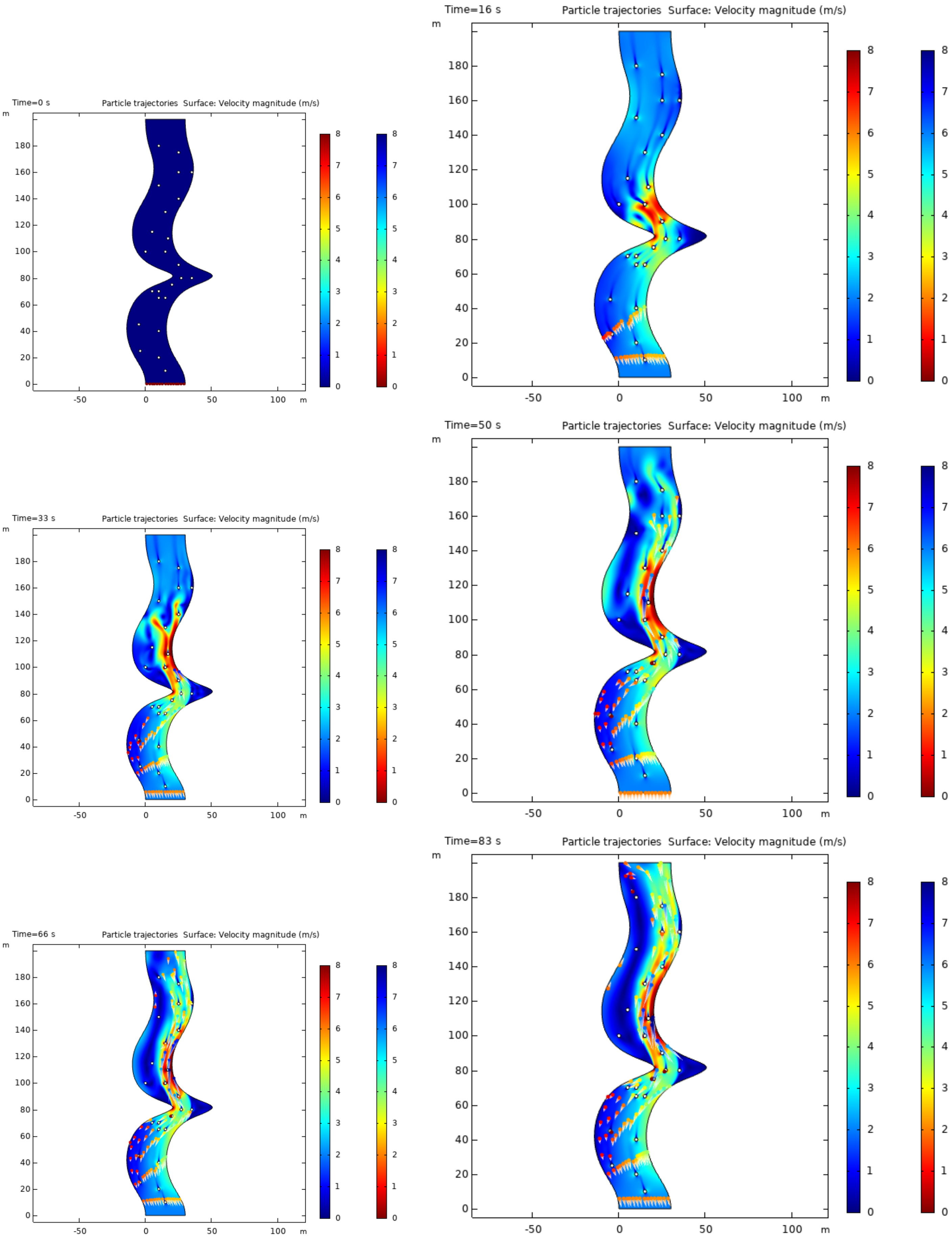



FIGURE 24. Particle trajectories superimposed over velocity magnitude evolution over 150 s, obstacles

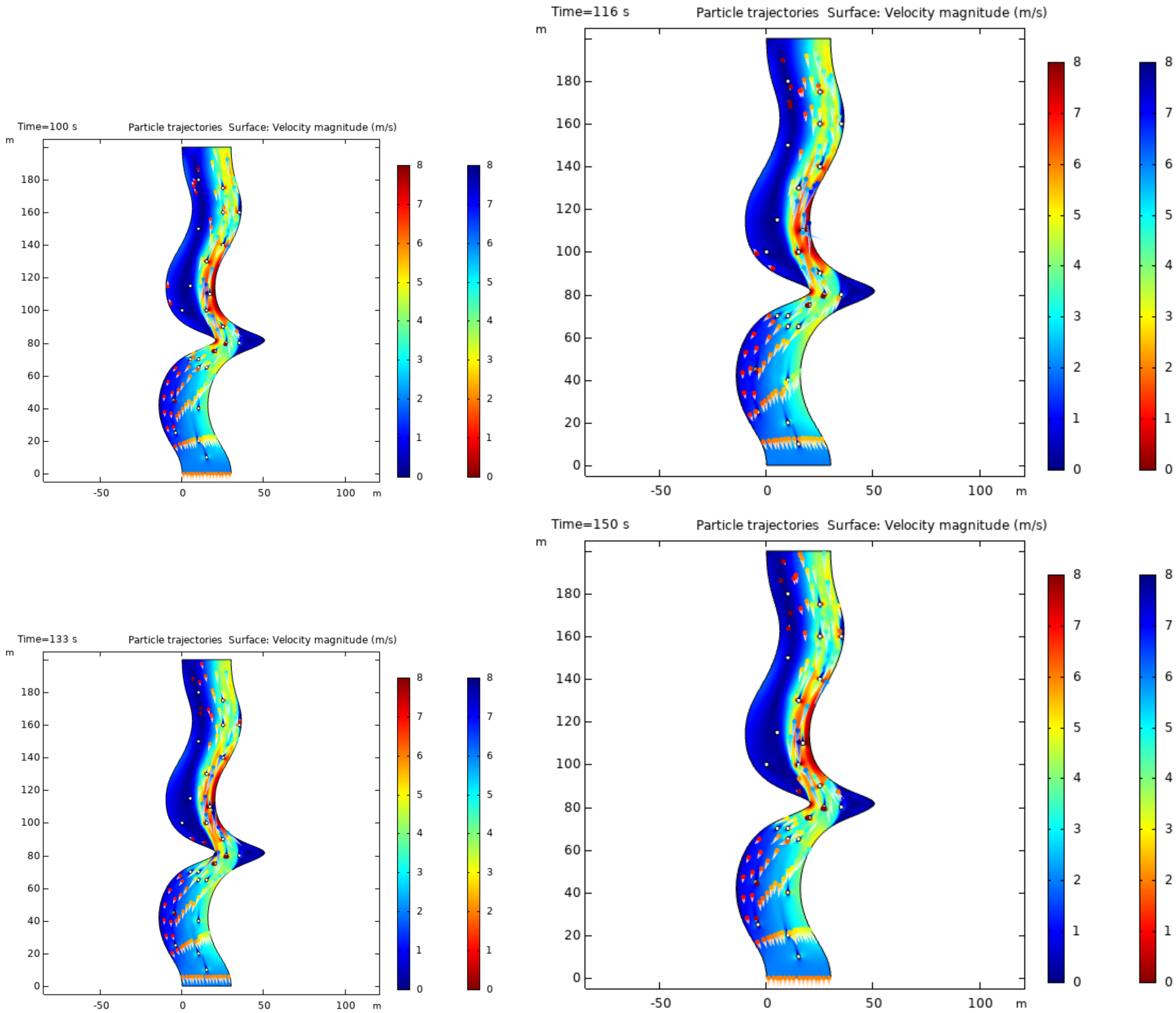



FIGURE 25. Particle trajectories superimposed over velocity magnitude evolution over 150 s, obstacles

## References


[1] Churuksaeva, V., & Starchenko, A. (2015) Mathematical Modeling of a River Stream Based on a Shallow Water Approach. *Procedia Computer Science*, vol. 66, pp. 200-209, doi:10.1016/j.procs.2015.11.024.

[2] Herrera-Díaz, I.E., et al. (2017) Light Particle Tracking Model for Simulating Bed Sediment Transport Load in River Areas. *Mathematical Problems in Engineering*, vol. 2017, pp. 1-15. doi:10.1155/2017/1679257.

[3] Yue, Z., et al. (2008) Two-Dimensional Coupled Mathematical Modeling of Fluvial Processes with Intense Sediment Transport and Rapid Bed Evolution. *Science in China Series G: Physics, Mechanics and Astronomy*, vol. 51(9), pp. 1427-1438, doi:10.1007/s11433-008-0135-1.

[4] Wetzel, R. G. (2001). Limnology: Lake and River Ecosystems (3rd ed.). San Diego, CA: Academic Press.



Department of Mathematics, CB 3250, University of North Carolina, Chapel Hill, NC, 27599

*Email address*: `ckofroth@live.unc.edu`