
# Bumpy Road Towards a Good LaTeX Visual Editor

## TUG 2023

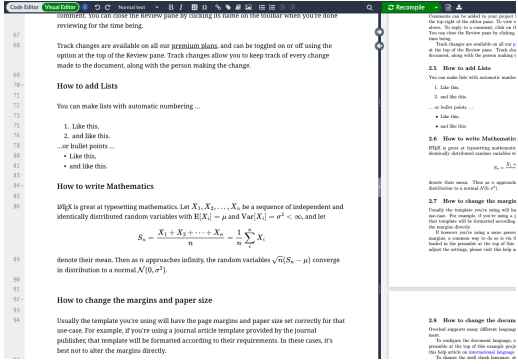
Ben Davies

# Collaboration


Overleaf

# Is There a Need?

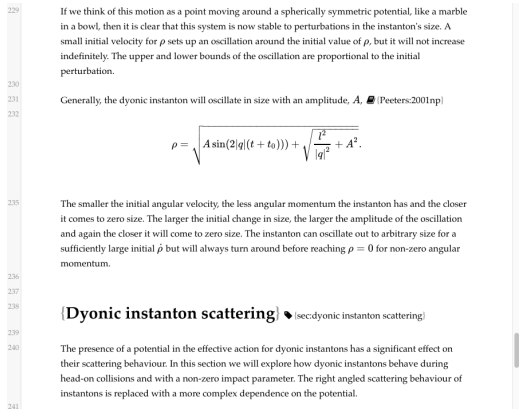
- ‘My [supervisor] doesn’t know LaTeX but I need to work with them’
- Proofreading tool whether it is you, an editor, a friend *etc.*
- You want to learn LaTeX




Overleaf

# Rich Text

- This used CodeMirror 5
- Introduced previews of maths and figures
- Some amount of code hiding
- But it didn't integrate well with Source and the Review features
- Core principle was to ensure code is always accessible



If we think of this motion as a point moving around a spherically symmetric potential, like a marble in a bowl, then it is clear that this system is now stable to perturbations in the instanton's size. A small initial velocity for $\rho$ sets up an oscillation around the initial value of $\rho$, but it will not increase indefinitely. The upper and lower bounds of the oscillation are proportional to the initial perturbation.

Generally, the dyonic instanton will oscillate in size with an amplitude, $A$, {Peeters:2001np}

$$\rho = \sqrt{A\sin(2|q|(t+t_0)) + \sqrt{\frac{L^2}{|q|^2} + A^2}}.$$

The smaller the initial angular velocity, the less angular momentum the instanton has and the closer it comes to zero size. The larger the initial change in size, the larger the amplitude of the oscillation and again the closer it will come to zero size. The instanton can oscillate out to arbitrary size for a sufficiently large initial $\dot{\rho}$ but will always turn around before reaching $\rho = 0$ for non-zero angular momentum.

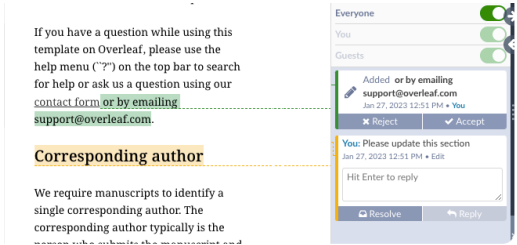
## {Dyonic instanton scattering} {sec:dyonic instanton scattering}

The presence of a potential in the effective action for dyonic instantons has a significant effect on their scattering behaviour. In this section we will explore how dyonic instantons behave during head-on collisions and with a non-zero impact parameter. The right angled scattering behaviour of instantons is replaced with a more complex dependence on the potential.

Overleaf

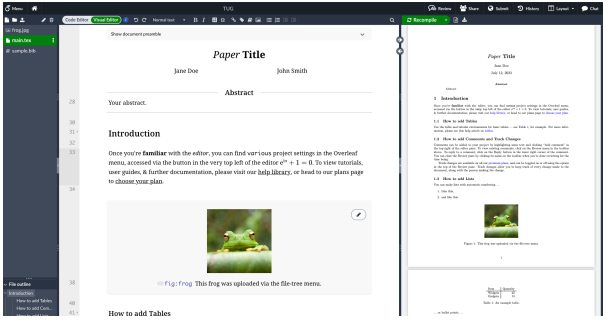
# Visual Editor

- Migrated to CodeMirror 6 (for both editors)
- Feature sharing more easily done such as Advanced Reference Search and Tracked changes
- Gave access to themes, keybindings and the same auto-complete
- Code editor with decorations
- Element-by-element design approach to improving the experience




Overleaf

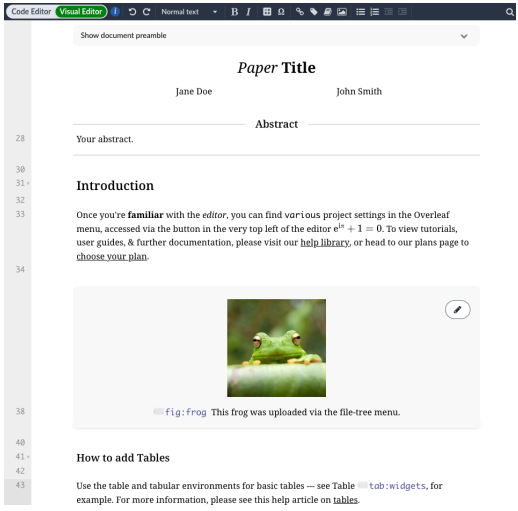
# Demo




Overleaf

# Things to think about

- Can we handle more variety efficiently?
- How much code should be hidden?
- What conventions/styles should we use?
- WYSIWYG conditioning
- Sharing behaviours/features across the editors



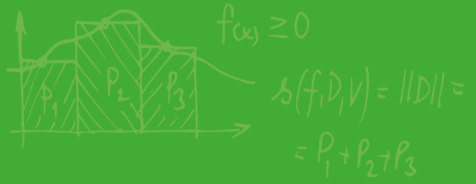

Overleaf

# Takeaways

- Providing parity to different code editors (CM6, CM5, Ace) is difficult
- Just because we know how we would write the LaTeX doesn't mean we know what the 'button' should do
- Providing different interfaces enhances user experience

# Thank you for listening!

# Any questions?




Overleaf