For

$$a = \frac{1000c}{\sqrt{b^2 + c} + b}.$$

we get

$$a \to \frac{1000\, c\, \varepsilon_r^{\cancel{3}\,2}}{\Big(\underbrace{\sqrt{b\varepsilon_r^4 + c\varepsilon_r^2}}_{O(\varepsilon_r^2)} + \underbrace{b\varepsilon_r}_{O(\varepsilon_r)}\Big)\cancel{\varepsilon_r}}$$

$$\to \frac{O(\varepsilon_r^2)}{O(\varepsilon_r)} = O(\varepsilon_r)$$

So $O(\varepsilon_r)$ here, which is a lot better. This is simply due to less operations in a way that the rounding errors can cancel.

In numbers using Numpy's single precision:

$$a = 1000\left(\frac{c}{\sqrt{b^2 + c} - b} - 2b\right) = 2.0000001100\boxed{59709}$$

$$a = \frac{1000c}{\sqrt{b^2 + c} + b}. = 2.0000001100\boxed{038816}$$