Analysis of algorithm.

In general using formula of $D\phi(i), D\psi^{(i)}$ etc. notations will become more difficult because here there are many steps in the algorithm and the derivatives of y also become complicated.

So a more realistic estimation of round off error has been done here using the fact that there are some inherent error carried out from the errors in the inputs plus some rounding error in each step....

there is error somewhere during making derivative of phi.

Algorithm A:

$$y = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

So $\frac{\partial\phi}{\partial a} = -\frac{-b\pm\sqrt{b^2-4ac}}{2a^2}$, $\frac{\partial\phi}{\partial b} = \frac{-1\pm\frac{1}{\sqrt{b^2-4ac}}}{2a}$, $\frac{\partial\phi}{\partial c} = \mp\frac{1}{\sqrt{b^2-4ac}}$.

The step of the algorithms are as follows:

$\phi(0) \coloneqq s \coloneqq b^2$

$\phi(1) \coloneqq t \coloneqq s - 4ac$

$\phi(2) \coloneqq u \coloneqq \sqrt{t}$

$\phi(3) \coloneqq v \coloneqq -b \pm u$

$\phi(4) \coloneqq w = v/(2 * a)$

The inherent error in w is $\epsilon_w^{(0.)} = \frac{\Delta^\circ w}{w} = \frac{\partial\phi}{\partial a}\epsilon_a + \frac{\partial\phi}{\partial b}\epsilon_b + \frac{\partial\phi}{\partial c}\epsilon_c = -\frac{1}{a}\epsilon_a + \frac{-1\pm\sqrt{b^2-4ac}}{-b\pm\sqrt{b^2-4ac}}\epsilon_b + \mp\frac{2a}{(-b\pm\sqrt{b^2-4ac})\sqrt{b^2-4ac}}\epsilon_c$.

Round off error in $\phi(2)$ is $\Delta u = \epsilon\sqrt{t} = \epsilon\sqrt{b^2 - 4ac}$ where ϵ is the floating point error in calculating the square root i.e. $fl(\sqrt{t}) = \sqrt{t}(1+\epsilon)$ with ϵ as $2\epsilon_b - \epsilon_a - \epsilon_c$.

So it will contribute to relative error in w as $\frac{\Delta u}{w} = \frac{2a\sqrt{b^2-4ac}}{-b\pm\sqrt{b^2-4ac}}\epsilon$.

$$\Longrightarrow \frac{\Delta u}{w} = \frac{2a\sqrt{b^2-4ac}(-b\mp\sqrt{b^2-4ac})}{4ac}\epsilon = \frac{-b\sqrt{b^2-4ac}\mp(b^2-4ac)}{2c}(2\epsilon_b - \epsilon_a - \epsilon_c).$$

So total rounding off error is

$$= -\frac{1}{a}\epsilon_a + \frac{-1+\sqrt{b^2-4ac}}{-b+\sqrt{b^2-4ac}}\epsilon_b - \frac{2a}{\left(-b+\sqrt{b^2-4ac}\right)\sqrt{b^2-4ac}}\epsilon_c - \frac{b\sqrt{b^2-4ac}+(b^2-4ac)}{2c}(2\epsilon_b - \epsilon_a - \epsilon_c).$$

where the $\pm$ sign is replaced by + sign only considering the analysis of one root of an algorithm.

This algorithm is not numerically stable because the round off error caused from $\sqrt{b^2-4ac}$ is very large compared to inherent error. Here amplitude factor |k|>>1.

Algorithm B:

$y = \frac{2c}{-b\pm\sqrt{b^2-4ac}}$

So $\frac{\partial\phi}{\partial a} = \frac{\pm 2c^2}{\sqrt{b^2-4ac}(-b\pm\sqrt{b^2-4ac})}$, $\quad \frac{\partial\phi}{\partial b} = \frac{2c}{\sqrt{b^2-4ac}(-b\pm\sqrt{b^2-4ac})}$, $\quad \frac{\partial\phi}{\partial c} = \frac{2}{-b+\sqrt{b^2-4ac}}$.

The step of the algorithms are as follows:

$\phi(0) := s := b^2$

$\phi(1) := t := s - 4ac$

$\phi(2) := u := \sqrt{t}$

$\phi(3) := v := -b \pm u$

$\phi(4) := w = (2 * c)/v$

The inherent error in w is $\epsilon_w^{(0.)} = \frac{\Delta^{\circ} w}{w} = \frac{\partial\phi}{\partial a}\epsilon_a + \frac{\partial\phi}{\partial b}\epsilon_b + \frac{\partial\phi}{\partial c}\epsilon_c = \frac{\pm c}{\sqrt{b^2-4ac}}\epsilon_a + \frac{1}{\sqrt{b^2-4ac}}\epsilon_b +$ $\mp\frac{a/c}{\sqrt{b^2-4ac}}\epsilon_c$.

Round off error in $\phi(2)$ is $\Delta u = \epsilon\sqrt{t} = \epsilon\sqrt{b^2-4ac}$ where ϵ is the floating point error in calculating the square root i.e. $fl(\sqrt{t}) = \sqrt{t}(1+\epsilon)$ with ϵ as $2\epsilon_b - \epsilon_a - \epsilon_c$.

Let define $\psi(u) = \frac{2c}{-b\pm u}$

So, it will contribute to relative error in w as $\frac{\frac{\partial\psi}{\partial u}\Delta u}{w} = \frac{-v}{(-b\pm u)^2}\sqrt{b^2-4ac}\epsilon$

$= -\frac{\sqrt{b^2-4ac}}{-b\pm\sqrt{b^2-4ac}}\epsilon = -\frac{\sqrt{b^2-4ac}}{-b\pm\sqrt{b^2-4ac}}(2\epsilon_b - \epsilon_a - \epsilon_c).$

So total rounding off error is

$$= \boldsymbol{\frac{c}{\sqrt{b^2-4ac}}\epsilon_a + \frac{1}{\sqrt{b^2-4ac}}\epsilon_b - \frac{a/c}{\sqrt{b^2-4ac}}\epsilon_c - \frac{\sqrt{b^2-4ac}}{-b+\sqrt{b^2-4ac}}(2\epsilon_b - \epsilon_a - \epsilon_c)}.$$

This algorithm is numerically stable because the round off error caused from $\sqrt{b^2-4ac}$ is very small compared to inherent error. Here amplitude factor |k| < 1.

Algorithm C:

$q = -\frac{1}{2}[b + sgn(b)\sqrt{b^2 - 4ac}]$.

So $\frac{\partial\phi}{\partial a} = \frac{bc}{\sqrt{b^2-4ac}}$; $\frac{\partial\phi}{\partial b} = -\frac{1}{2}\left(\frac{b+\sqrt{b^2-4ac}}{\sqrt{b^2-4ac}}\right), \frac{\partial\phi}{\partial c} = \frac{a}{\sqrt{b^2-4ac}}$.

The step of the algorithms are as follows:

$\phi(0) := s := b^2$

$\phi(1) := t := s - 4ac$

$\phi(2) := u := \sqrt{t}$

$\phi(3) := v := -\frac{1}{2}(b + u)$

The inherent error in v is $\frac{\partial\phi}{\partial a}\epsilon_a + \frac{\partial\phi}{\partial b}\epsilon_b + \frac{\partial\phi}{\partial c}\epsilon_c = \frac{bc}{\sqrt{b^2-4ac}}\epsilon_a - \frac{1}{2}\left(\frac{b+\sqrt{b^2-4ac}}{\sqrt{b^2-4ac}}\right)\epsilon_b + \frac{a}{\sqrt{b^2-4ac}}\epsilon_c$.

Round off error in $\phi(2)$ is $\Delta u = \epsilon\sqrt{t} = \epsilon\sqrt{b^2 - 4ac}$ where ϵ is the floating point error in calculating the square root i.e. $fl(\sqrt{t}) = \sqrt{t}(1 + \epsilon)$ with ϵ as $2\epsilon_b - \epsilon_a - \epsilon_c$.

So it will contribute relative error in w as $\frac{\Delta u}{v} = -\frac{2\sqrt{b^2-4ac}}{b+\sqrt{b^2-4ac}}\epsilon = -\frac{2\sqrt{b^2-4ac}}{b+\sqrt{b^2-4ac}}(2\epsilon_b - \epsilon_a - \epsilon_c)$.

So total rounding off error is $\frac{bc}{\sqrt{b^2-4ac}}\epsilon_a - \frac{1}{2}\left(\frac{b+\sqrt{b^2-4ac}}{\sqrt{b^2-4ac}}\right)\epsilon_b + \frac{a}{\sqrt{b^2-4ac}}\epsilon_c - \frac{2\sqrt{b^2-4ac}}{b+\sqrt{b^2-4ac}}(2\epsilon_b - \epsilon_a - \epsilon_c)$.

This algorithm is not numerically stable because the round off error caused from $\sqrt{b^2 - 4ac}$ is very large compared to inherent error. Here amplitude factor |k|>>1.

Finally the round off error in algorithm B is the lowest than others. So algorithm B is numerically trustworthy. This can be verified by analusing the coefficients of the corresponding expressions.

<u>Conclusion:</u>

Algorithm B is more numerically trust worthy.

Also **algorithm B** is more numerically stable.

Hence algorithm B is the better algorithm from both view.