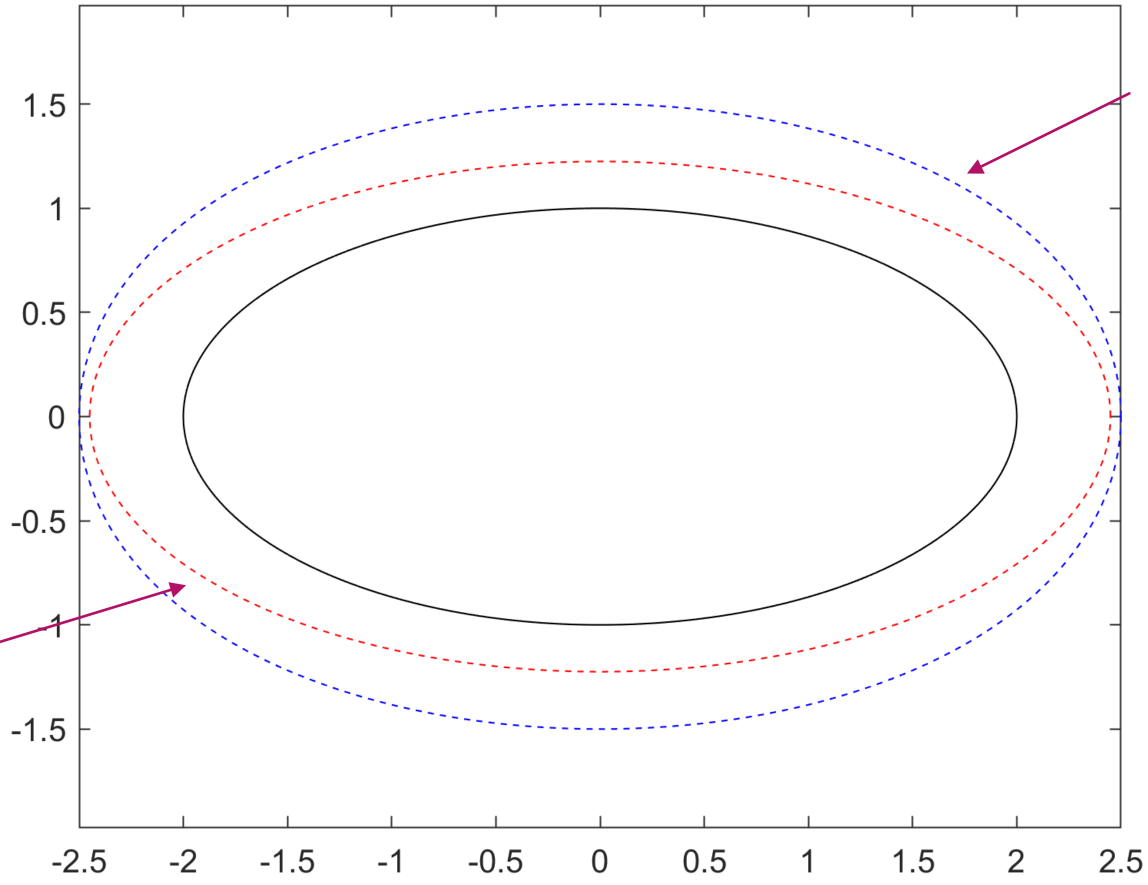
Exact and Convergent Iterative Methods to Compute the Orthogonal Point-to-Ellipse Distance

SIYU GUO

HUNAN UNIVERSITY, CHANGSHA, CHINA

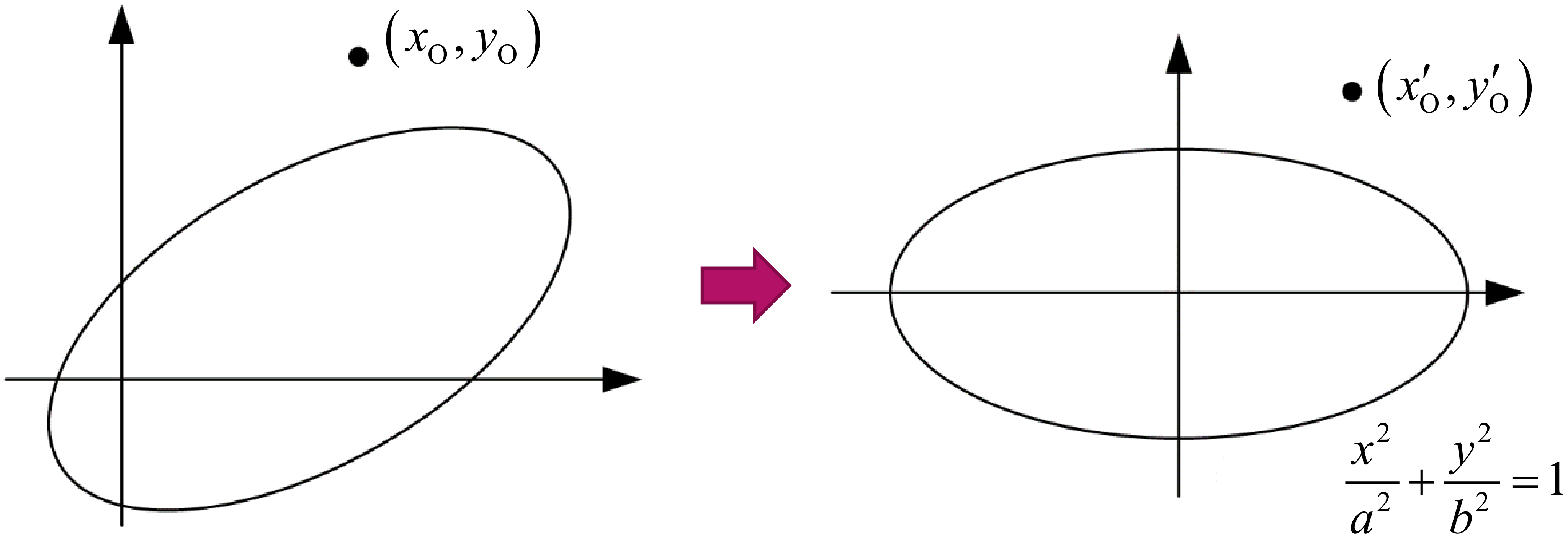
Orthogonal distance and algebraic distance

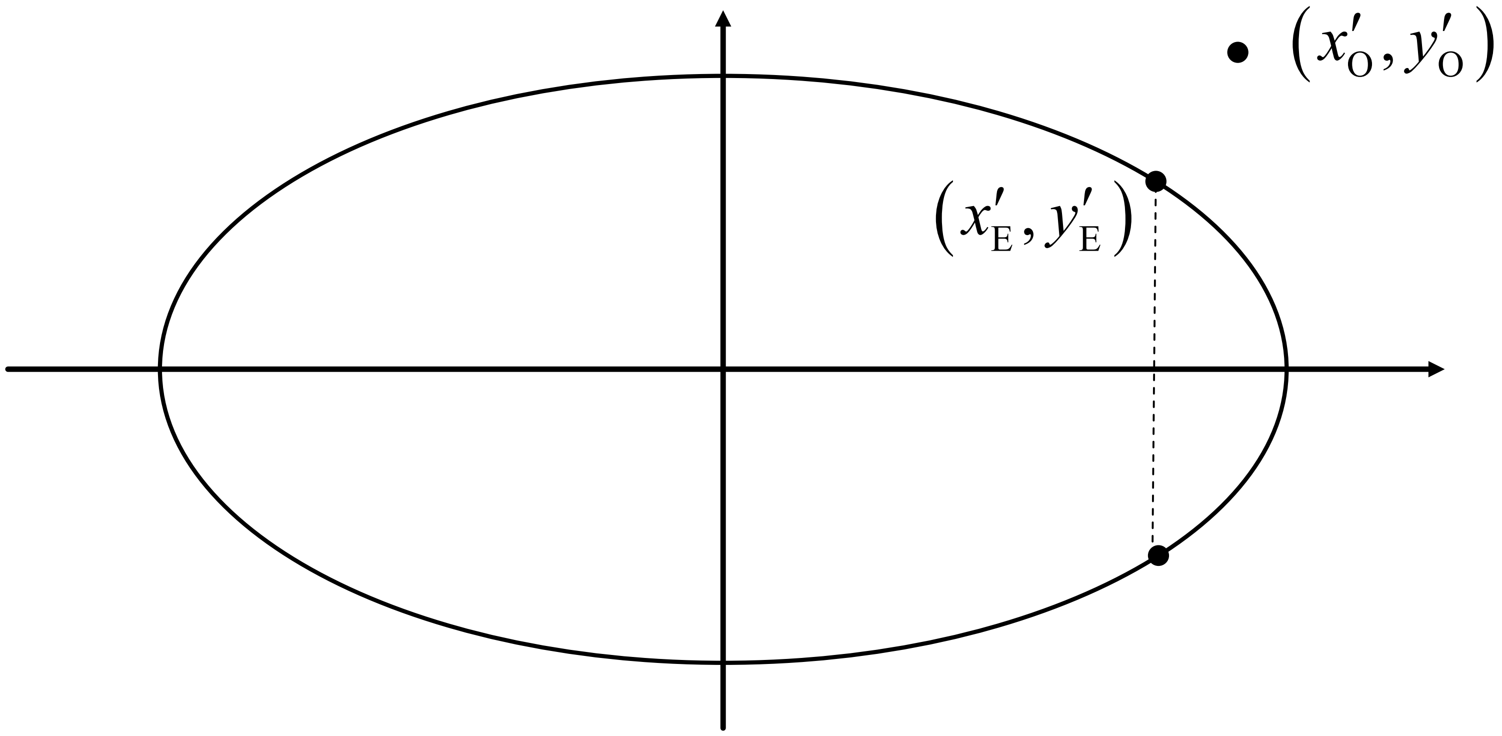


Iso-distance curve by orthogonal distance

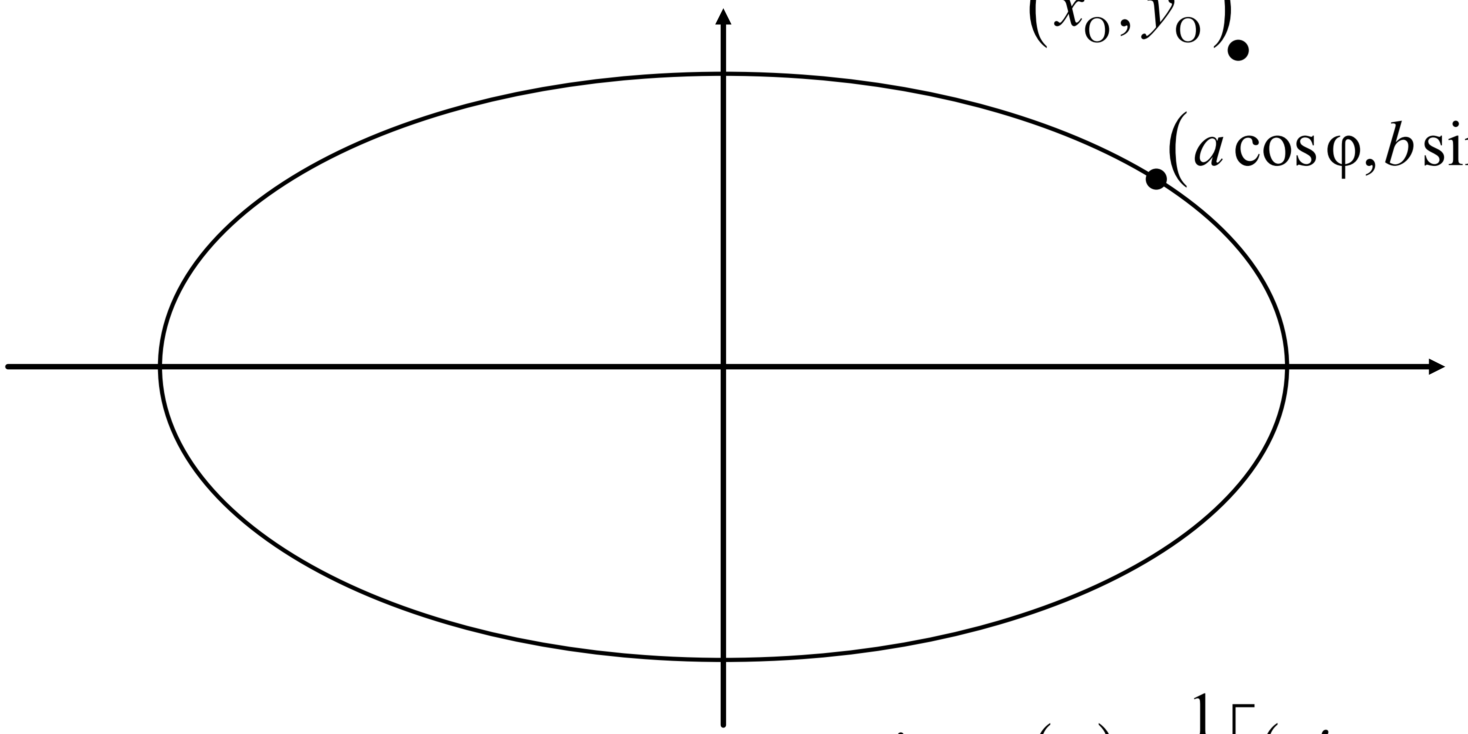
Iso-distance curve by algbraic distance

Problem formulation




(x'_O, y'_O)
(x'_E, y'_E)

$$(x'_O, y'_O)$$



$$\left(a\cos\varphi, b\sin\varphi\right), 0 \le \varphi \le \frac{\pi}{2}$$

$$\min_{\varphi\in[0,\pi/2]} D(\varphi) = \frac{1}{2}\left[\left(x'_O - a\cos\varphi\right)^2 + \left(y'_O - b\sin\varphi\right)^2\right]$$

$$\alpha_4 s^4 + \alpha_3 s^3 + \alpha_2 s^2 + \alpha_1 s + \alpha_0 = 0$$

$$s = \sin\varphi, 0 \le s \le 1$$

$$\begin{cases} \alpha_4 = -d^2 \\ \alpha_3 = -\alpha_1 = -2y'_O bd \\ \alpha_2 = d^2 - \left(a^2 x'^2_O + b^2 y'^2_O\right) \\ \alpha_0 = b^2 y'^2_O \end{cases}$$

$$d = a^2 - b^2$$

The exact algorithm

$$\begin{cases} s_{1,2} = -\dfrac{\alpha_3}{4\alpha_4} + \dfrac{1}{2}\left(R \pm \sqrt{S+T}\right) \\ s_{3,4} = -\dfrac{\alpha_3}{4\alpha_4} - \dfrac{1}{2}\left(R \pm \sqrt{S-T}\right) \end{cases}$$

$$R = \sqrt{\frac{\alpha_3^2}{4\alpha_4^2} - \frac{\alpha_2}{\alpha_4} + y_{\mathrm{r}}}$$

$$S = \begin{cases} \dfrac{3\alpha_3^2}{4\alpha_4^2} - \dfrac{2\alpha_2}{\alpha_4} & R = 0 \\ \dfrac{3\alpha_3^2}{4\alpha_4^2} - \dfrac{2\alpha_2}{\alpha_4} - R^2 & R \neq 0 \end{cases}$$

$$T = \begin{cases} 2\sqrt{y_{\mathrm{r}}^2 - \dfrac{4\alpha_0}{\alpha_4}} & R = 0 \\ \dfrac{1}{4R\alpha_4^3}\left(4\alpha_2\alpha_3\alpha_4 - 8\alpha_1\alpha_4^2 - \alpha_3^3\right) & R \neq 0 \end{cases}$$

…

$$\begin{cases} s_{1,2} = -\dfrac{\alpha_3}{4\alpha_4} + \dfrac{1}{2}\left(R \pm \sqrt{S+T}\right) \\ s_{3,4} = -\dfrac{\alpha_3}{4\alpha_4} - \dfrac{1}{2}\left(R \pm \sqrt{S-T}\right) \end{cases}$$

- Computationally expensive
- Numeric errors during the computation may fail to give the desired real root. Can be overcome by extra examinations at the expense of running speed.

The convergent iterative algorithm

$$\alpha_4 s^4 + \alpha_3 s^3 + \alpha_2 s^2 + \alpha_1 s + \alpha_0 = 0$$

$$s_{k+1} = \frac{\left(\left(3\alpha_1 s_k + 2\alpha_2\right) s_k + \alpha_3\right) s_k^2 - \alpha_5}{\left(\left(4\alpha_1 s_k + 3\alpha_2\right) s_k + 2\alpha_3\right) s_k + \alpha_4}, k = 0, 1, \ldots$$

How to choose the initial solution for Newton's method to guarantee the convergence to the expected root?

$$\alpha_4 s^4 + \alpha_3 s^3 + \alpha_2 s^2 + \alpha_1 s + \alpha_0 = 0$$

$$s_{k+1} = \frac{\left(\left(3\alpha_1 s_k + 2\alpha_2\right) s_k + \alpha_3\right) s_k^2 - \alpha_5}{\left(\left(4\alpha_1 s_k + 3\alpha_2\right) s_k + 2\alpha_3\right) s_k + \alpha_4}, k = 0,1,\ldots$$

- The equation has only one single root on $[0,1]$.
- With $s_0 = 1$ as the initial solution, the Newton's iteration converges to the root, and the order of convergence is 2.

Experimental results

- 2001×2001 grid points over the square region of [-10,10]×[-10,10]. Parameters of the ellipse is $X_C = Y_C = 0$, $a = 1$, $b = 0.5$, $\theta = \pi/6$.

Algorithm	ARW01	EA	CA
Exe. Time (s)	2.463	12.054	0.967
r_{SD}	0.55	0.41	0.41
r_{AF}	0.45	0	0
d_{MO}	2.2×10^{-6}	0.49	2.2×10^{-6}

- r_{SD} – Proportion of being the most accurate.
- r_{AF} – Proportion of failure.
- d_{MO} – Distance lag to the best.

Experimental results

- 100 randomly generated ellipses, each with 100-1000 randomly generated points.

Algorithm	ARW01	EA	CA
Exe. Time (s)	0.008	0.258	0.017
r_{SD}	0.47	0.32	0.27
r_{AF}	0.01	0	0
d_{MO}	9.5×10^{-6}	6.3×10^{-6}	6.3×10^{-6}

Conclusions

- ARW01 is shown in the experiments to produce more accurate solutions, but the lead is minute.
- ARW01 suffers of divergence.
- The convergence of CA is theoretically proven and experimentally verified.
- CA and ARW01 are much faster than EA. Clock errors considered, CA should be regarded as the fastest among the three algorithms.

Your visit to Hunan University is warmly welcome!