Billiards and chaos

In this scenario, we want to model the chaotic trajectory of a billiard ball (approximated as a point-like particle) within some arbitrarily-defined borders. The problem we're ultimately interested in solving is when $r(\phi) - r_b(\phi) = 0$ for some particle at position (r, ϕ), moving with unit-magnitude velocity in a two-dimensional plane. That is to say, we want to know where its position in (r, ϕ) space overlaps with some boundary located at $r_b(\phi)$. For two-dimensional motion, we have two equations in Cartesian coordinates that indicate when a particle has reached a boundary,

$$x_i + v_x \Delta t - x_b = 0, \text{ and} \tag{1}$$

$$y_i + v_y \Delta t - y_b = 0. \tag{2}$$

However, we also know that v_x and v_y are related by $\tan\theta_i = v_y/v_x$. So in principle, before reaching the boundary, the particle is constrained to move along a straight line with slope $|\vec{v}|$, which makes an angle θ_i to the x-axis. Therefore, there is no need for the precise evaluation of $v\Delta t$. Rather, the evaluation of our boundary condition becomes a simple midpoint problem of the form,

$$r_{\text{mid}}(\phi_{\text{mid}}) - r_b(\phi) = 0, \text{ where} \tag{3}$$

$$\begin{aligned} r_{\text{mid}}(\phi_{\text{mid}}) &= \sqrt{x_{\text{mid}}^2 + y_{\text{mid}}^2} \\ &= \sqrt{\left(\frac{1}{2}(x_i + x_N)\right)^2 + \left(\frac{1}{2}(y_i + y_N)\right)^2}. \end{aligned} \tag{4}$$

In this case, we have,

$$x_N = x_i + N \times \cos(\theta_i), \; y_N = y_i + N \times \sin(\theta_i), \text{ and,} \tag{5}$$

$$\phi_{\text{mid}} = \text{atan2}\left(\frac{y_N + y_i}{x_N + x_i}\right). \tag{6}$$

This relationship exploits the fact that the midpoint $(x_{\text{mid}}, y_{\text{mid}})$ is constrained to be on the line between (x_i, y_i) and (x_N, y_N) by definition. So the precise knowledge of $v\Delta t$ is unnecessary, as we require only that the term N is sufficiently large so that x_N and y_N "overshoot" the boundary,

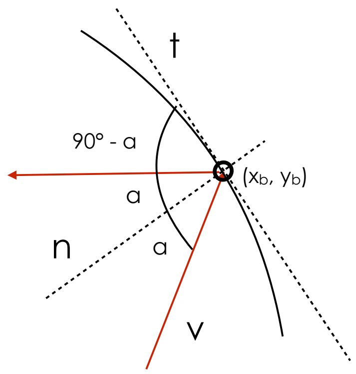


Figure 1: Diagram of the relation between the tangent $\hat{t}$, normal $\hat{n}$, and velocity $\vec{v}$ at a boundary point (x_b, y_b).

i.e. $r_N(\phi_N) > r_b(\phi_N)$ initially. The problem can then be solved using root finding via bisection, with the points (x_i, y_i) and (x_N, y_N) serving as the initial bisection domain, and the midpoint $(r_{\text{mid}}, \phi_{\text{mid}})$ recomputed iteratively, as necessary.

Recalculating velocities

For each given point (x_b, y_b) along a boundary, a unique tangent line $\vec{t}$ exists which intersects only with the point (x_b, y_b), and no others along the boundary. Supposing we can find a normal vector $\hat{n}$ which is perpendicular to the tangent line $\vec{t}$ at the boundary (x_b, y_b), the component of the particle's velocity perpendicular to the boundary at (x_b, y_b) is given as,

$$\vec{v}_\perp = \vec{v} \cdot \hat{n} = \vec{v} \cdot \frac{\vec{n}}{|\vec{n}|}. \tag{7}$$

Correspondingly, the component of the velocity parallel to the boundary is given as,

$$\vec{v}_\parallel = \vec{v} - \vec{v}_\perp. \tag{8}$$

Furthermore, the magnitude of the velocity projection onto the normal is related to the incident velocity vector by $\cos(\alpha)$, so one can calculate the angle between the particle trajectory and the line tangent to the boundary as $\pi/2 - \alpha$, or $\pi/2 - \cos^{-1}(|\vec{v}_\perp|/|\vec{v}|)$, as shown in Figure 1.

Of course, for the case of a circle, finding the unit normal vector $\hat{n}$ is trivial, since the tangent line is always perpendicular to the radius. However, for the case of an ellipsoid, we are required first to compute $m_t = dy/dx|_{x=x_b, y=y_b}$

at a given point, such that the slope of the normal line is $m_n = -1/(dy/dx)$. From this point, one can construct a normal vector of arbitrary length, requiring only that it intersect with the tangent at the boundary (x_b, y_b).

Therefore, for the ellipsoid boundary, we can calculate the slope of the tangent line $m_{\vec{t}} = dy/dx$ at a boundary point in (r, ϕ) in polar coordinates using the relation,

$$m_{\vec{t}} = \frac{dy}{dx} = \frac{\frac{dr}{d\phi} \sin\phi + r\cos\phi}{\frac{dr}{d\phi}\cos\phi - r\sin\phi} \tag{9}$$

The parametric equation for the ellipsoid shape is defined in terms of the angle ϕ as follows,

$$r(\phi) = 1 + 0.16\cos(2\phi) + 0.1\sin(\phi). \tag{10}$$

So, taking the derivative of this radial function with respect to ϕ, we have,

$$\frac{dr}{d\phi} = -2 \times 0.16\sin(2\phi) + 0.1\cos(\phi), \tag{11}$$

from which the slope of the line normal to the tangent can be computed as $m_{\vec{n}} = -1/m_{\vec{t}}$, requiring also that it intersects the boundary at (x_b, y_b). The unit-length normal line can then be computed as $\hat{n} = \vec{n}/|\vec{n}|$. In order to compute the normal vector in MATLAB, code of the following form was used:

```
%Create a line along the normal.
dydx       = tangentslope(phi);
xvals_int  = linspace(xval-0.7,xval+1.5,100);
yvals_int  = (-1/dydx)*(xvals_int-xval)+yval;
normperp   = [xvals_int(end)-xvals_int(1);yvals_int(end)-
    yvals_int(1)];

%Convert to a unit vector.
normperp   = normperp / norm(normperp);
```

An example of the first few trajectories of the particle, showing the tangent and normal lines at each (x_b, y_b), is shown in Figure 2 for the ellipsoid boundary with initial conditions $(x_i, y_i, \theta_i) = (0.2, 0.6, 0.1)$.

Particle trajectories

The first test of the code involved simulating a series of 50 bounces for a given set of boundary conditions and initial parameters (x_i, y_i, θ_i), with the

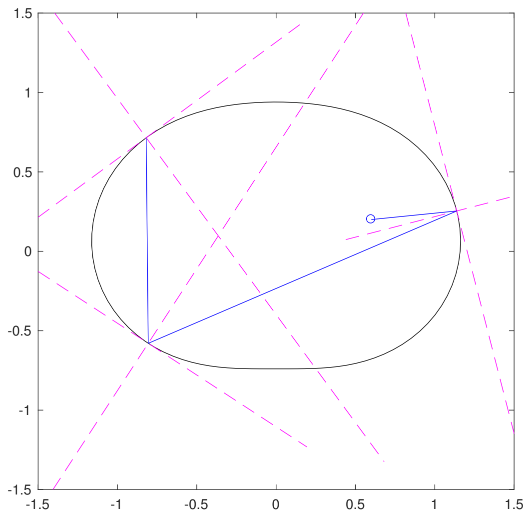


Figure 2: First few bounces of the particle in the ellipsoid boundary with initial conditions $(x_i, y_i, \theta_i) = (0.2, 0.6, 0.1)$ (solid lines), showing the tangent and normal lines at each point (dotted lines).

bisection method used to find the intersection of the particle's trajectory with the boundary. The velocity of the particle is then recalculated at each boundary collision according to the prescription in Section 1. The simulations were performed using two sets of parameters – namely, sets of "nominal" parameters, and "nudged" parameters for which the starting y-position is modified by $\epsilon = 10^{-7}$.

Nominal parameters

For each boundary type, two sets of initial parameters were chosen, namely $(x_i, y_i, \theta_i) = (0.2, 0.6, 0.1)$, and $(x_i, y_i, \theta_i) = (0.1, 0.0, 1.5)$, denoted the "nominal" parameters. The distributions in Figure 3 show the trajectory of the particle within each boundary for 50 bounces with each set of starting parameters.

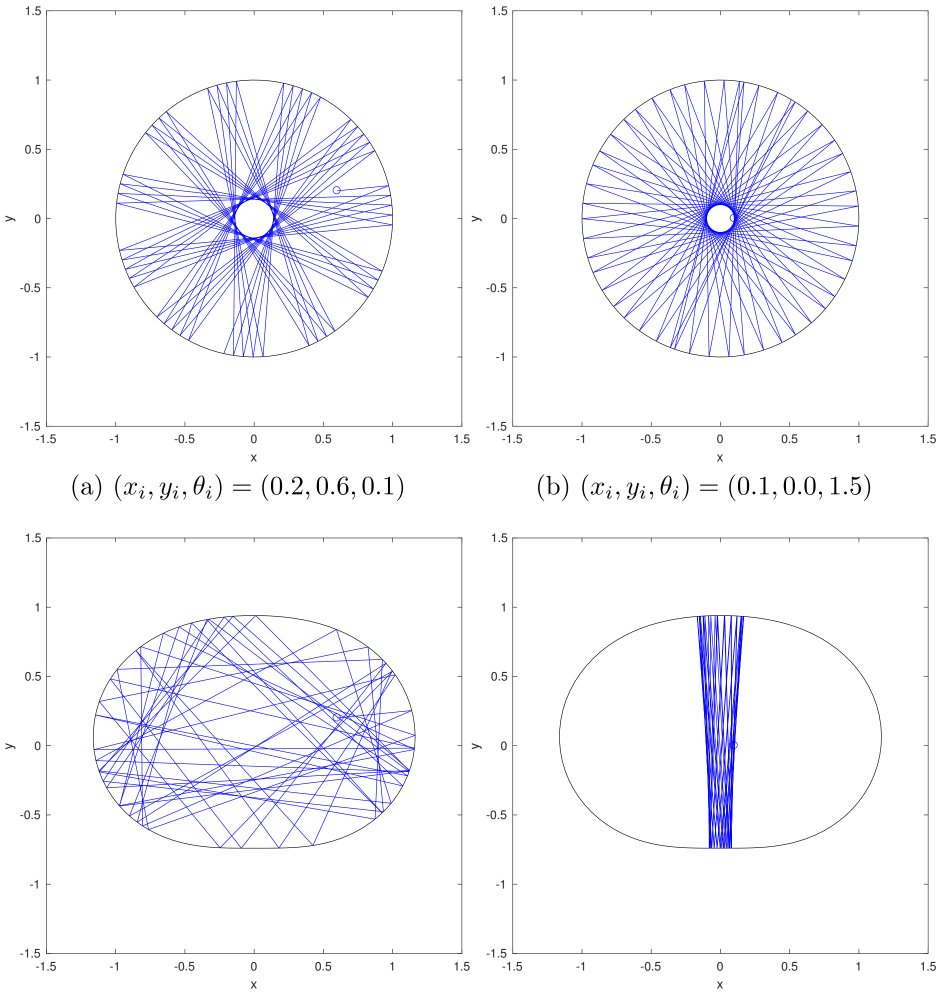
Considering the circular boundary, the two sets of initial conditions produce comparable results in trajectory, in that both sets of parameters produce "star-like" patterns where the trajectory consistently rebounds around a central focus at (0,0). These results are not totally unexpected; because the curvature of a circle is constant at every (x_b, y_b), the angle of incidence of each subsequent bounce will be the same. Therefore, the magnitude of the parallel component of the velocity will remain constant (a point reflected when looking at the Poincaré sections of Section 3), leading the particle to reflect back and forth across opposite ends of the boundary, with the end-points travelling around the border in a circular fashion. The main difference between the two plots lies primarily in the starting velocity; because the second set of parameters has a much smaller initial $\vec{v}_{||}$, the separation between bounces in the ϕ direction is much more narrow, leading to a finer pattern, and smaller orbital radius around (0,0).

Considering the ellipsoid boundary, on the other hand, it is evident that the two sets of initial conditions produce drastically different outputs. It is worth noting that for the trajectory starting at $(x_i, y_i, \theta_i) = (0.1, 0.0, 1.5)$, the angle of incidence is $\sim 90°$, which leads to distinct behaviour for the ellipsoid boundary, as shown in the bottom right of Figure 3. In particular, the particle's movement remains bounded to the narrow area in the middle of the ellipsoid, since the boundary is effectively flat in this region, and the initial velocity is practically all contained in the perpendicular component. Therefore, the component of the velocity parallel to the tangent never becomes large enough to "kick" the particle out of this region. Conversely, the first set of initial conditions sees the particle's trajectory spanning nearly the entirety of each boundary in ϕ, owing to the fact that the trajectory intersects with regions of a high degree of curvature multiple times. Therefore, the path appears much more random, although some systematic trends the particle's path can be found, as discussed further in Section 3.

"Nudged" parameters

The trajectory distributions were also reproduced using initial parameters which were "nudged" in the y-direction by a small *epsilon* of $\epsilon = 10^{-7}$. It is evident from the resultant distributions that the change in the path taken by the particle in the circular boundary is negligible, as shown in Figure 4. With a circular boundary, due to the constant nature of the curvature, a small change of ϵ in the y direction should manifest only as a constant, trivially small change in path. Likewise, for the ellipsoid boundary with initial conditions $(x_i, y_i, \theta_i) = (0.1, 0.0 + \epsilon, 1.5)$, no meaningful change is observed in the trajectory of the particle. Indeed, comparing vectors of the (x_b, y_b) pairs of the nominal and "nudged" parameters showed no meaningful deviations at each boundary collision point.

However, a very visible change in the trajectory of the particle for the ellipsoid boundary is seen for initial parameters of $(x_i, y_i, \theta_i) = (0.2, 0.6 + \epsilon, 0.1)$. Due to the smallness of the ϵ, the tolerance for the bisection algorithm was set to 10^{-8} to limit the possibility that the differences in boundary finding were the result of inadequate precision in the root finding, rather than real physics effects. Initially, a relative difference of 0.5% in either the x or y



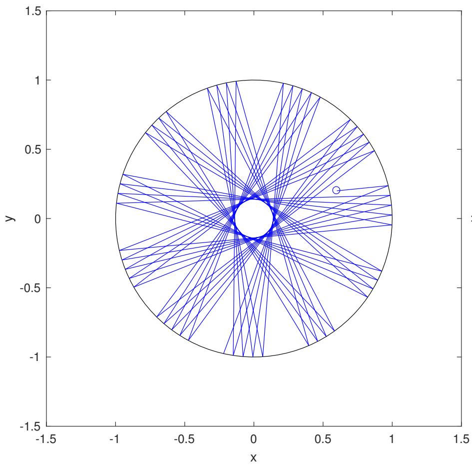
(a) $(x_i, y_i, \theta_i) = (0.2, 0.6, 0.1)$

(b) $(x_i, y_i, \theta_i) = (0.1, 0.0, 1.5)$

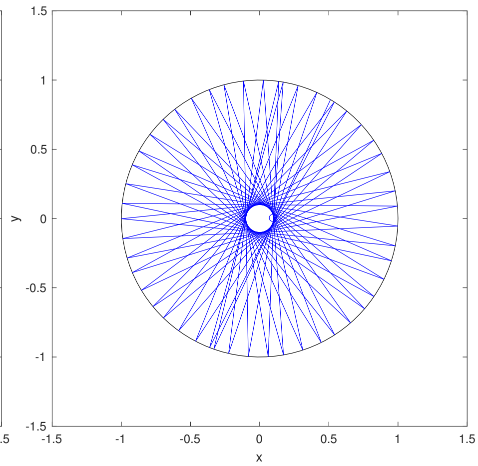
(c) $(x_i, y_i, \theta_i) = (0.2, 0.6, 0.1)$

(d) $(x_i, y_i, \theta_i) = (0.1, 0.0, 1.5)$

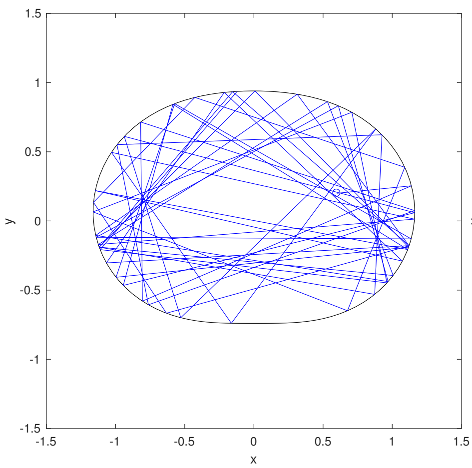
Figure 3: Boundary kinematic plots with nominal parameters in y.



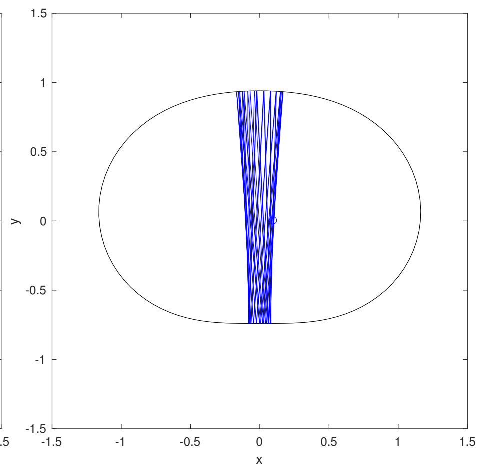
(a) $(x_i, y_i, \theta_i) = (0.2, 0.6 + \epsilon, 0.1)$



(b) $(x_i, y_i, \theta_i) = (0.1, 0.0 + \epsilon, 1.5)$



(c) $(x_i, y_i, \theta_i) = (0.2, 0.6 + \epsilon, 0.1)$



(d) $(x_i, y_i, \theta_i) = (0.1, 0.0 + \epsilon, 1.5)$

Figure 4: Boundary kinematic plots with the “nudged” parameter in y.

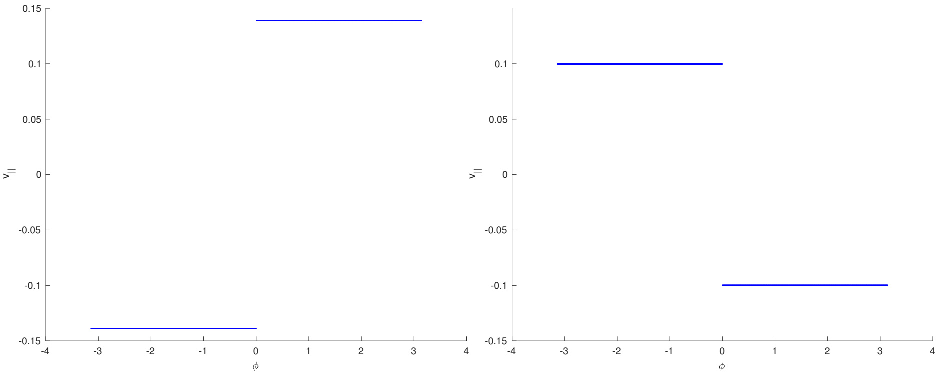
position was considered a “significant” deviation in path. Considering this threshold, Table 1 presents a list of significant deviations in path within 50 bounces. It was found that the first significant deviation in trajectory, according to this threshold, occurred after the 15th bounce, with subsequent steps deviating further in x and y.

In order to understand whether the deviations were simply due to inaccuracies in finding the contact points and angles, the threshold for a “meaningful” deviation was reduced to ϵ and instead treated as an absolute (rather than a relative) error. With this choice of threshold, as shown in Table 2, a deviation of $\mathcal{O}(\epsilon)$ is found immediately at the second bounce, with differences in Δx and Δy increasing with subsequent steps. However, it is worth noting that the deviations Δx_b and Δx_y accrue gradually and systematically from $\mathcal{O}(10^{-8})$ to $\mathcal{O}(1)$, rather than fluctuating or appearing randomly, and that in general $\mathcal{O}(\Delta x_b) \sim \mathcal{O}(\Delta y_b)$ at each bounce. Due to this systematic behaviour, it appears that the different end results are actually the effect of small, cumulative differences in the trajectory of the particle, rather than some fluctuation in the root finding or velocity recalculation.

Poincaré projections

A series of Poincaré projections were also produced for each configuration of boundary and initial conditions. These projections were produced by plotting the component of the velocity parallel to the boundary ($v_{\parallel}$) against the axial angle of the particle (ϕ) at a given boundary collision point (x_b, y_b). Because the vector $\vec{v}_{\parallel}$ in principle has both magnitude and direction, the calculation of the scalar $v_{\parallel}$ was made by projecting the particle's velocity vector onto the normalized tangent vector at (x_b, y_b), i.e. $v_{\parallel} = \vec{v} \cdot \hat{t} = \vec{v} \cdot \vec{t}/|\vec{t}|$.

The resultant Poincaré projections for the circular and ellipsoid boundaries are given in Figures 5 and 6, respectively, for two choices of initial conditions, $(x_i, y_i, \theta_i) = (0.2, 0.6, 0.1)$ and $(x_i, y_i, \theta_i) = (0.1, 0.0, 1.5)$. It is evident that the projections for the circular barrier are reflective of the symmetry of the boundary; at each boundary point, the magnitude of the parallel velocity component $|\vec{v}_{\parallel}|$ remains the same. As previously mentioned, this behaviour is not unexpected; because the curvature of a circle is the same at each (x_b, y_b), the angle of incidence of each subsequent bounce will be the same, meaning that $|\vec{v}_{\parallel}|$ will be constant. Furthermore, the pattern shown in the projections suggests that the particle tends to bounce back and forth into opposite quadrants, a point which is reaffirmed when looking at its trajectory in x and y. Moreover, the particle will tend to return to the starting point after a certain number of bounces, due to the symmetry of the bound-

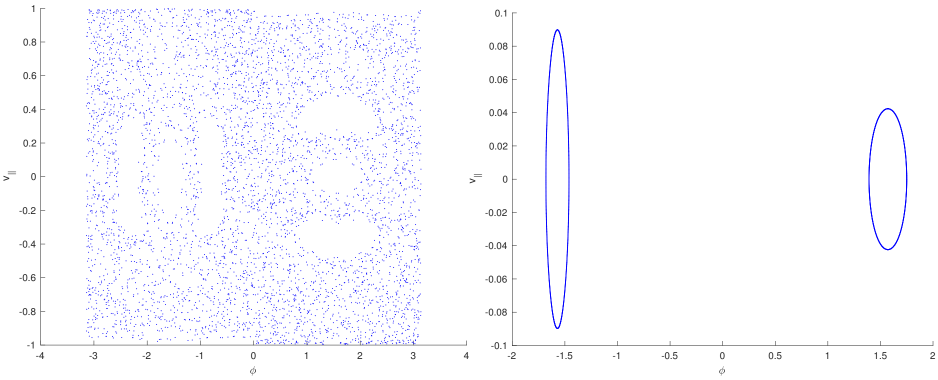


(a) $(x_i, y_i, \theta_i) = (0.2, 0.6, 0.1)$ (b) $(x_i, y_i, \theta_i) = (0.1, 0.0, 1.5)$

Figure 5: Poincaré projections for the circular boundary at different initial conditions.

ary. Consequently, multiple reflections will create a "star"-shaped pattern will emerge around the central focus, i.e. the point (0,0), and the parallel component of the velocity will remain constant.

For the projections using the ellipsoid boundary, with initial conditions $(x_i, y_i, \theta_i) = (0.2, 0.6, 0.1)$, there is a greater deal of variance in $|\vec{v}_{\parallel}|$, as is expected from the the particle's more varied trajectory. Interestingly, regions of essentially zero population are visible in this distribution around $\phi \sim \pm 1.5$, meaning that the particle's travel is not entirely random or chaotic, but has some periodic trends, which is potentially due to the symmetry of the boundary conditions. A more non-uniform boundary could potentially result in a Poincaré distribution resembling Gaussian noise. Conversely, the second set of conditions at $(x_i, y_i, \theta_i) = (0.1, 0.0, 1.5)$ produces a very stable projection, with the only population occurring within a pair of ovals around $\phi \sim \pm 1.6$. This behaviour is reflective of the limited range in ϕ that the particle travels with these initial conditions, as the initial trajectory is very close to $\pi/2$, so the trajectory remains practically all in the perpendicular direction after each bounce. Consequently, the full range of possible $v_{\parallel}$ values are spanned only within a small region of ϕ.



(a) $(x_i, y_i, \theta_i) = (0.2, 0.6, 0.1)$

(b) $(x_i, y_i, \theta_i) = (0.1, 0.0, 1.5)$

Figure 6: Poincaré projections for the ellipsoid boundary at different initial conditions.

Tables of x_b, y_b deviations for nominal and "nudged" parameters

Step	x_b	y_b	x_b (nudged)	y_b (nudged)	$\Delta x/x_b$ (%)	$\Delta y/y_b$ (%)
15	-1.146111	-1.146714	0.222150	0.219288	-0.05	1.29
17	-0.541876	-0.586305	0.859037	0.842562	-8.20	1.92
20	-0.655122	-0.943171	0.812723	0.605249	-43.97	25.53
22	1.119984	0.869829	-0.185127	0.675026	22.34	-464.63
23	0.022629	-0.141835	0.939884	-0.739662	726.78	178.70
25	1.004529	1.140648	-0.396817	-0.116309	13.55	-70.69
26	0.536920	-1.112419	0.860750	0.336298	307.19	60.93
28	1.160394	1.048304	0.004751	0.467161	9.66	9732.18
29	-1.011118	-0.855841	0.523048	0.686668	-15.36	31.28
31	0.815359	0.993184	0.717870	-0.411168	21.81	157.28
33	-1.012759	-1.124948	0.520777	0.300831	-11.08	42.23
34	1.161875	-0.233881	0.027765	-0.737715	120.13	2756.99
36	0.168120	-0.963701	0.933452	-0.445504	673.22	147.73
37	1.087231	-0.120507	-0.263470	0.936673	111.08	-455.51
39	-0.235392	-0.802085	0.926888	0.727363	-240.74	21.53
40	1.085217	-0.441184	-0.267584	-0.717803	140.65	-168.25
41	-1.158595	0.589611	0.143586	0.841252	-150.89	485.89
42	1.162683	0.314237	0.075822	-0.733240	72.97	1067.05
44	0.987991	-0.382703	0.553442	-0.726405	138.74	231.25
45	0.275753	-0.103634	-0.735795	0.937547	137.58	-227.42
46	-0.774269	0.343059	0.746178	-0.730748	-144.31	197.93
47	0.113219	0.457428	-0.739860	0.885076	304.02	-219.63
48	0.915450	-0.339365	0.633520	-0.731098	137.07	215.40
50	-0.195200	0.757636	0.931103	-0.608670	-488.13	165.37

Table 1: Table of significant deviations for the nominal and "nudged" starting parameters with $(x_i, y_i, \theta_i) = (0.2, 0.6, 0.1)$ for the ellipsoid boundary.

Step	x_b	y_b	x_b (nudged)	y_b (nudged)	Δx	Δy
3	-1.121739	-1.121739	0.310450	0.310450	0.00000011	0.00000030
4	0.385639	0.385638	-0.726037	-0.726038	0.00000129	0.00000016
5	1.156946	1.156946	0.157896	0.157895	0.00000011	0.00000085
6	0.025892	0.025893	0.939848	0.939848	0.00000127	0.00000001
7	-1.156305	-1.156305	0.162895	0.162895	0.00000009	0.00000062
8	-0.362088	-0.362086	-0.728803	-0.728803	0.00000250	0.00000027
9	1.111948	1.111948	0.337531	0.337532	0.00000052	0.00000134
10	-1.161959	-1.161959	0.099315	0.099312	0.00000014	0.00000315
11	1.043569	1.043575	-0.341358	-0.341347	0.00000651	0.00001018
12	0.112917	0.112881	0.937083	0.937085	0.00003655	0.00000191
13	-1.041844	-1.041851	-0.344039	-0.344030	0.00000628	0.00000972
14	1.109203	1.109232	0.344606	0.344532	0.00002917	0.00007407
15	-0.825229	-0.825483	-0.566437	-0.566261	0.00025471	0.00017639
16	-0.716743	-0.716206	0.780927	0.781227	0.00053774	0.00030048
17	1.128910	1.128668	-0.158261	-0.159034	0.00024144	0.00077378
18	-1.158127	-1.157820	0.147900	0.150610	0.00030684	0.00271006
19	1.152733	1.150953	-0.057853	-0.068163	0.00177973	0.01031000
20	-1.042140	-1.018808	0.477138	0.512268	0.02333138	0.03513064
21	-0.040370	-0.146295	-0.739998	-0.739619	0.10592548	0.00037838
22	1.005959	0.877777	0.530080	0.668198	0.12818197	0.13811756
23	-1.052597	-0.534883	-0.326902	-0.697656	0.51771417	0.37075464
24	0.402913	-0.928927	0.898618	0.620094	1.33184009	0.27852407
25	0.994613	1.162198	-0.409398	0.035120	0.16758509	0.44451816
26	-1.162616	-1.007611	0.049204	-0.392795	0.15500485	0.44199860
27	0.956849	-0.356101	0.590308	0.908422	1.31294999	0.31811417
28	0.463628	1.046320	-0.713734	-0.337028	0.58269118	0.37670527
29	-0.831844	-1.006712	0.705589	0.529063	0.17486748	0.17652508
30	0.079836	0.055391	-0.739965	-0.739992	0.02444533	0.00002698
31	0.930170	1.054543	0.618826	0.456725	0.12437268	0.16210108
32	-0.924133	-1.162231	-0.485988	0.035989	0.23809820	0.52197743
33	-0.168383	1.107148	0.933431	-0.218861	1.27553070	1.15229151
34	0.914317	-0.339464	-0.495188	0.911529	1.25378112	1.40671746
35	-0.582120	-1.074327	0.844202	-0.288813	0.49220640	1.13301505
36	-0.725863	1.019336	-0.625673	-0.376980	1.74519888	0.24869347
37	0.799072	0.552169	0.729470	0.855401	0.24690234	0.12593113
38	0.225734	-1.141107	-0.737999	-0.114453	1.36684111	0.62354653
39	-0.526580	1.114674	0.864246	-0.199678	1.64125387	1.06392459
40	-0.194724	-0.266204	-0.738856	0.923036	0.07148064	1.66189192
41	0.228775	-1.098528	0.927644	-0.239127	1.32730297	1.16677099
42	0.270587	0.934805	-0.736077	-0.475627	0.66421785	0.26044971
43	0.134273	0.808647	0.935857	0.722713	0.67437459	0.21314476
44	-0.216288	-1.131574	-0.738297	0.279493	0.91528520	1.01778981
45	-0.456752	-0.137521	0.885259	-0.739701	0.31923081	1.62495937
46	0.193084	1.060048	-0.738892	0.447211	0.86696412	1.18610332
47	0.723956	-1.138403	0.776855	-0.125111	1.86235830	0.90196616
48	-0.519250	1.057223	-0.701618	0.452131	1.57647273	1.15374865
49	-0.831200	-0.117908	0.706079	-0.739836	0.71329213	1.44591502
50	1.067175	-1.124130	-0.301920	0.303325	2.19130468	0.60524490

Table 2: Absolute deviations in x and y between the nominal and "nudged" parameters for the ellipsoid with starting parameters $(x_i, y_i, \theta_i) = (0.2, 0.6, 0.1)$.