AAS 11-480

COMPARISONS BETWEEN NEWTON-RAPHSON AND BROYDEN'S METHODS FOR TRAJECTORY DESIGN PROBLEMS

Matthew M. Berry*

Broyden's method, a generalized-secant method for root-finding, was recently added as an option in the STK/Astrogator maneuver planning and trajectory design software module. The software previously used a Newton-Raphson approach with numerical partials to solve shooting problems. In this paper, the two methods are compared for a wide variety of problems, including station-keeping, orbit transfers, and interplanetary trajectories. For most use-cases, Broyden's method has a faster performance than Newton-Raphson.

INTRODUCTION

Root-finding algorithms, such as Newton-Raphson and Broyden's methods, are useful in solving trajectory design problems. These algorithms are used to adjust the problem's independent variables, such as the time, direction, and magnitude of maneuvers, to achieve certain desired values of dependent variables, such as the final orbital characteristics. The STK/Astrogator maneuver planning and trajectory design software module has used a Newton-Raphson algorithm to solve root-finding problems since the software was first released. Recently, an option was added to the software to use Broyden's method instead. This paper describes the two algorithms, explains how they are implemented in STK/Astrogator, and compares using them to solve a variety of trajectory design problems.

ROOT-FINDING METHODS

Root-finding methods solve problems of the type

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) = \mathbf{0}, \tag{1}$$

where $\mathbf{x}$ is a vector of independent variables and $\mathbf{y}$ is a vector of dependent variables. If the desired values, $\mathbf{y_d}$, of the dependent variables are not zero, the problem can be written as

$$\tilde{\mathbf{y}} = \mathbf{f}(\mathbf{x}) - \mathbf{y_d} = \mathbf{0}. \tag{2}$$

To solve for $\mathbf{x}$, methods typically start from some initial guess, $\mathbf{x_0}$, compute the function $\mathbf{f}(\mathbf{x})$, and compare the resulting $\mathbf{y}$ values to the desired values. The methods then compute a step to take in the independent variables so that the dependent variables will move closer to the desired values.

* Sr. Astrodynamics Engineer, Analytical Graphics, Inc. 220 Valley Creek Blvd. Exton, PA 19341. mberry@agi.com.

Each cycle of setting independent variables and computing the function is called an evaluation of the problem. Certain algorithms perform several evaluations to determine the next step to take; each cycle of determining the next step is called an iteration of the algorithm. In trajectory design problems, where evaluations of the function may require significant computation time, methods that reduce the total number of evaluations are desirable.

Newton-Raphson Method

The Newton-Raphson method uses the first derivative of the function to determine the step to take in the independent variables. At each iteration, the **x** values for the next iteration are chosen such that the root would be reached if **f**(**x**) were linear. In single-variable problems, the method is[1]

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \tag{3}$$

where x_k represents the value of the independent variable on the k^{th} iteration, and $f'(x)$ is the first derivative. Equation (3) is written as if the desired values are zero for simplicity; the method can be modified if they are not.

If the first derivative cannot be computed analytically, it can be computed numerically by using a small perturbation δx. In this case the method is

$$x_{k+1} = x_k - \frac{f(x_k)}{f(x_k + \delta x) - f(x_k)} \delta x . \tag{4}$$

In multi-variable problems, the Newton-Raphson method is[2]

$$\mathbf{x_{k+1}} = \mathbf{x_k} - \mathbf{J_n^{-1}}\left(\mathbf{f}(\mathbf{x_k}) - \mathbf{y_d}\right), \tag{5}$$

where **J** is the Jacobian matrix. The Jacobian is comprised of the partial derivatives,

$$\mathbf{J} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}, \tag{6}$$

where n is the number of independent variables and m is the number of dependent variables. If **J** is not square, the pseudo-inverse of **J** is used in Eq. (5).

To compute **J** numerically, the function is evaluated with the independent variables perturbed. The i^{th} column of **J** is given by

$$\mathbf{J_i} = \frac{1}{\delta x_i}\left(\mathbf{f}(\mathbf{x} + \boldsymbol{\delta}\mathbf{x_i}) - \mathbf{f}(\mathbf{x})\right), \tag{7}$$

where the scalar δx_i is the perturbation of the i^{th} independent variable and the vector $\boldsymbol{\delta x_i}$ is

$$\boldsymbol{\delta}\mathbf{x_i} = \mathbf{x} + \begin{Bmatrix} 0 \\ \vdots \\ 0 \\ \delta x_i \\ 0 \\ \vdots \\ 0 \end{Bmatrix}. \tag{8}$$

Computing **J** numerically requires n evaluations of **f**(**x**). After Eq. (5) is used to find the step to take in the independent variables, a final evaluation is performed to see if the dependent variables are within tolerance of their desired values. If they are not, another iteration is performed. Thus, for a problem with n independent variables, each iteration requires n+1 evaluations. Because an evaluation is also required before the first iteration to compute the initial values of the dependent variables, a solution in k iterations requires $(n+1)k + 1$ evaluations.

To help the method find a solution, the step size found by Eq. (5), $\mathbf{\Delta x} = \mathbf{x_{k+1}} - \mathbf{x_k}$, may be limited by a maximum step size allowed for each variable, $\Delta x_{i\,\max}$. If any of the variables are over its limit the entire step $\mathbf{\Delta x}$ is divided by the ratio of $\Delta x_{i\,\max} / \Delta x_i$. Because the Newton-Raphson method is using the local Jacobian to search for a solution, using a maximum step size is helpful in non-linear problems where the initial guess is not close to the solution. The maximum step limit helps prevent the method from over-shooting the solution.

Broyden's Method

In the secant method, instead of using the first derivative, successive iteration values are used to estimate the location of the root. In single-variable problems, the algorithm is[3]

$$x_{k+1} = x_k - f(x_k)\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}. \tag{9}$$

The secant method is similar to using Newton-Raphson with numerical derivatives, Eq. (4), but instead of using a small perturbation to estimate the slope, the method uses values over a wider range of the independent value.

Broyden's method is a generalization of the secant method for multi-variable problems.[4] The method is similar to the multi-variable Newton-Raphson method, Eq. (5), but uses an estimate of the Jacobian instead of evaluating the Jacobian on every iteration. The Jacobian estimate is

$$\mathbf{J_k} = \mathbf{J_{k-1}} + \frac{\mathbf{f}(\mathbf{x_k}) - \mathbf{f}(\mathbf{x_{k-1}}) - \mathbf{J_{k-1}}(\mathbf{x_k} - \mathbf{x_{k-1}})}{\left\|\mathbf{x_k} - \mathbf{x_{k-1}}\right\|^2} \left(\mathbf{x_k} - \mathbf{x_{k-1}}\right)^{\mathrm{T}}. \tag{10}$$

On the first iteration, the initial value of the Jacobian is found using numerical derivatives as it is in Newton-Raphson. Because of the initial evaluation, with n independent variables this iteration requires n+2 iterations. On subsequent iterations Eq. (10) is used, only requiring a single evaluation of **f**(**x**). Thus, a solution in k iterations of a problem with n independent variables requires $k + n + 1$ evaluations. The number of evaluations per iteration with Broyden's method is less than it is with the Newton-Raphson method. However, Broyden's method may require more iterations to converge to a solution than Newton-Raphson because Eq. (10) is just an estimate of the actual Jacobian values computed in Eq. (7).

IMPLEMENTATION IN STK/ASTROGATOR

To model trajectories, users of STK/Astrogator create a Mission Control Sequence (MCS), which consists of trajectory segments that the users configure.[5,6] There are various types of trajectory segments: for example, propagate segments (which numerically propagate the trajectory until some stopping condition is met), impulsive maneuver segments (which add a ΔV in a specified reference frame), and finite maneuver segments (which numerically propagate the trajectory with a thrust acceleration given by a user-configured engine model).

To solve trajectory design problems in STK/Astrogator, users create target sequences consisting of segments that comprise the problem. The segments within a target sequence expose independent variables chosen by the user, such as the duration of a propagate segment or the ΔV magnitude of an impulsive maneuver. Users can also define segment results, which are values computed at a segment's end state such as eccentricity and inclination, to act as the dependent variables in targeting problems.

In addition to the segments, a target sequence also has a series of targeting profiles, which contain algorithms for solving the targeting problem. A profile can modify the independent variables of the segments in the target sequence, run those segments, and evaluate the segment results. Profiles using different algorithms can be chosen based on the desired goals of the problem. An optimization profile is available for problems that have a cost function as well as inequality or equality constraints. A differential corrector profile is available for problems that only have equality constraints, i.e., problems that can be described by Eq. (2).

In differential corrector profiles, the user can define the independent variables, the dependent variables, properties of the variables such as perturbations, maximum step size, desired values and tolerances, and the root-finding algorithm used. Multiple profiles might be used to solve a given problem, with profiles using the final solution from the previous profile as their initial guesses. The profiles might have different values for perturbations, maximum step size, desired value and tolerances, and may even use different independent and dependent variables. The multiple profile approach allows the user to create rough targeting problems with one set of variables that are easy to solve and then move into fine targeting problems that start with a good initial guess.

When STK/Astrogator was initially released, the differential corrector profile used the Newton-Raphson method, Eq. (5). To evaluate the function $\mathbf{f}(\mathbf{x})$ in the algorithm, the STK/Astrogator code sets the segments' independent variables to the current $\mathbf{x}$ values, runs the segments by numerically propagating the trajectory and accounting for maneuvers, and then calculates the dependent variable values, $\mathbf{y}$. The evaluation at the start of the method, in which the independent variables are equal to the initial guess, is referred to as the "initial run" by STK/Astrogator. To compute the Jacobian, evaluations are performed with each independent variable perturbed; STK/Astrogator refers to these computations as "perturbation runs". The final evaluation of an iteration, during which the independent variables use the new values calculated in Eq. (5), is referred to as the "nominal run" of the iteration.

Broyden's method has been implemented in STK/Astrogator as part of the version 10.0 release by adding an option to the differential corrector profile that allows the user to choose between the Newton-Raphson method and Broyden's method. When the profile runs, the first iteration always consists of the initial run, the n perturbation runs to compute $\mathbf{J}$, and the nominal run.

If the first iteration does not converge, the STK/Astrogator code checks to see which method is selected. If the Newton-Raphson method is selected, perturbation runs are performed to compute the Jacobian using Eq. (7). If Broyden's method is selected, the Jacobian is updated using Eq. (10). Once the Jacobian is calculated, the iteration proceeds the same way no matter which option is selected: the Jacobian is inverted, Eq. (5) is evaluated to find new values of the independent variables, **x**, and the nominal run of the iteration is performed with those **x** values.

In targeting problems with multiple independent variables but only a single dependent variable, the Jacobian is a column vector, initially found using numerical partials in the first iteration. Because the step taken in the independent variables is in the same direction as that column vector, updates to **J** through Broyden's Eq. (10) will also be in the same direction. If Eq. (10) was used on every iteration, and the solution was not along that line, the method would never converge. Therefore, when there are multiple independent variables but only one dependent variable, the STK/Astrogator code re-computes the Jacobian with perturbation runs whenever the solution starts to diverge. Divergence is detected whenever the dependent variable is further from its desired value than it was on the previous iteration.

TEST CASES

The examples below compare the Newton-Raphson method and Broyden's method in a variety of problems representative of typical STK/Astrogator use cases. The tests were performed using a development build of STK 10.0. Code changes that would affect these results are not expected between now and when version 10.0 is released (anticipated in early 2012).

Earth to Moon

The first test case is the problem of computing a trajectory that launches from Earth and ends at capture around the Moon, at a specified altitude and inclination, after a specified time of flight.[6] In STK/Astrogator, the target sequence for this problem has four segments: a launch segment into low earth orbit, a coast segment in low earth orbit, an impulsive maneuver segment performing the trans-lunar injection (TLI) burn, and a propagate segment to the Moon. Table 1 describes these segments and their independent and dependent variables.

Table 1: Description of segments in Earth to Moon example

Segment	Independent variables	Initial guess	Description
Launch	Launch Epoch	1/1/2020 12:00:00 UTC	Launches from 28.6° lat, -80.6° lon to circular orbit at 300 km altitude
Coast	Duration	45 minutes	Propagates until duration has passed
TLI burn	ΔV magnitude	3.14 km/sec	Performs ΔV in the velocity direction
Propagate to Moon	None		Propagates either until reaching lunar periapsis, or the time of flight has passed

The targeting problem is divided into three targeting profiles. In the first, the independent variables are the launch time and coast duration and the dependent variables are the difference between the spacecraft's and the Moon's final right ascension and declination. In this profile, the Propagate to Moon segment always propagates until the time of flight has passed. In the second profile, the ΔV magnitude is added as a third independent variable, and the three dependent variables are the time of flight and the two coordinates of the trajectory's B-plane. In this profile, the Propagate to Moon segment always propagates until reaching lunar periapsis. In the final profile, the three independent variables are the same as the second profile's and the dependent variables are the time of flight, the final lunar altitude, and the final lunar inclination. Again, the Propagate to Moon segment is set to propagate until reaching lunar periapsis. The perturbation values, maximum step size, desired values and tolerances used in the profiles are given in Table 2.

Table 2: Targeting profiles values used in Earth to Moon example

Profile	Independent variables			Dependent variables		
	Name	Perturbation	Max step	Name	Desired value	Tolerance
RA Dec	Launch Epoch	5,000 sec	15,000 sec	Delta declination	0°	0.1°
	Coast duration	100 sec	600 sec	Delta right ascension	0°	0.1°
B-plane	Launch Epoch	100 sec	10,000 sec	B dot R	6000 km	0.1 km
	Coast duration	10 sec	300 sec	B dot T	0 km	0.1 km
	ΔV magnitude	1e-4 km/sec	0.01 km/sec	Time of flight	5 days	0.1 sec
Altitude and inclination	Launch Epoch	10 sec	1000 sec	Lunar altitude	250 km	0.01 km
	Coast duration	1 sec	100 sec	Lunar inclination	90°	0.01°
	ΔV magnitude	1e-5 km/sec	0.01 km/sec	Time of flight	5 days	0.1 sec

Table 3 shows the number of iterations and evaluations used running the profiles with Newton-Raphson and Broyden's method. The table shows that, overall, Broyden's method is 40% faster than Newton-Raphson. The two methods find trajectories that are equivalent to one another – both are within the user-specified tolerances of the desired final conditions.

Table 3: Iterations and evaluations needed in Earth to Moon example

Profile	Newton-Raphson method		Broyden's method	
	Iterations	Evaluations	Iterations	Evaluations
RA Dec	5	16	7	10
B-plane	3	13	5	9
Altitude and inclination	3	13	2	6
Total		42		25

Interplanetary

The second test case is an interplanetary example replicating the New Horizons mission to Pluto.[7] The mission launched from Earth in January of 2006, and performed a Jupiter flyby on its way to Pluto. Segments modeling the mission are shown in Table 4.

Table 4: Description of segments in interplanetary example

Segment	Independent variables	Initial guess	Description
Launch	Launch epoch	1/19/ 2006 19:00:00.000 UTC	Launches from 28.6° lat, -80.6° lon to circular orbit at 300 km altitude
Coast	Duration	30 min	Propagates until duration has passed
Escape burn	ΔV magnitude	10 km/sec	Performs ΔV in the velocity direction
To Jupiter	None		Propagates to Jupiter periapsis
To Pluto	None		Propagates to Pluto periapsis

Three targeting profiles are used to solve the problem; they are described in Table 5. The first profile targets the launch epoch, coast duration, and ΔV magnitude so that the C3 energy and right ascension and declination of the outgoing asymptote after the escape burn put the spacecraft on a trajectory towards Jupiter. The necessary desired values of C3 energy, right ascension, and declination were computed in a separate process with the STK/Astrogator software. The second profile refines the launch epoch and coast duration with B-plane targeting so that the Jupiter flyby is the correct distance from Jupiter. The third profile is a final refinement of the launch epoch, coast duration, and escape burn magnitude so that after the Jupiter flyby the spacecraft is on a trajectory towards Pluto.

Table 6 shows the number of iterations and evaluations used by the methods to solve the interplanetary test case. The table shows that, overall, Broyden's method is 57% faster than Newton-Raphson for this case. The greatest difference is in the third profile, where the larger number of iterations needed by both methods - relative to the first and second profiles - enhances the advantage of Broyden's method.

Table 5: Targeting profiles values used in interplanetary example

Profile	Independent variables			Dependent variables		
	Name	Perturbation	Max step	Name	Desired value	Tolerance
Jupiter target vector	Launch epoch	600 sec	3600 sec	C3 energy	157.69462653 km^2/sec^2	1e-007 km^2/sec^2
	Coast duration	60 sec	600 sec	Outgoing asymptote declination	-8.9°	0.1°
	ΔV magnitude	0.001 km/sec	1 km/sec	Outgoing asymptote right ascension	209.857°	0.1°
Jupiter B-plane	Launch epoch	60 sec	3600 sec	B dot R	0 km	10 km
	Coast duration	6 sec	600 sec	B dot T	2.3e6 km	10 km
Pluto target vector	Launch epoch	1 sec	3600 sec	C3 energy	344.44150063 km^2/sec^2	1e-007 km^2/sec^2
	Coast duration	1 sec	600 sec	Outgoing asymptote declination	-18.56°	0.1°
	ΔV magnitude	1e-6 km/sec	0.1 km/sec	Outgoing asymptote right ascension	246.775°	0.1°

Table 6: Iterations and evaluations needed in interplanetary example

Profile	Newton-Raphson method		Broyden's method	
	Iterations	Evaluations	Iterations	Evaluations
Jupiter target vector	4	17	5	9
Target capture	2	7	2	5
Lower periapsis	9	37	8	12
Total		61		26

Libration Point

The third test case is an MCS replicating the WMAP mission[8], which is used to compare the methods in libration point trajectory problems. The WMAP mission is in a halo orbit about the Sun-Earth-Moon L2 point. The MCS for this problem consists of three target sequences, described in Table 7. The targeting profiles used by the target sequences are described in Table 8.

Table 7: Description of segments in libration point example

Segment	Independent variables	Initial guess	Description
Target on B-plane			
Launch	None		Launches at 8/11/2000 00:27:05.990 UTC from 28.6° lat, -80.6° lon to a circular orbit at 188.2 km altitude
Coast	None		Coasts in low-Earth parking orbit for a fixed duration
Third stage burn	None		Performs a burn in the velocity direction to achieve a C3 energy of -2.6 km^2/sec^2
Propagate to perigee1	None		Propagates to the next perigee
P1 burn	ΔV magnitude	0.02 km/sec	Performs ΔV in the velocity direction
Propagate to perigee2	None		Propagates to the next perigee
P2 burn	ΔV magnitude	.01 km/sec	Performs ΔV in the velocity direction
Propagate to perigee3	None		Propagates to the next perigee
Return	None		Propagation stops here after targeting
To periselene	None		Propagates to lunar periapsis
Target Pf maneuver			
Pf maneuver	ΔV magnitude	5.22e-5 km/sec	Performs ΔV in the velocity direction
To periselene	None		Propagates to lunar periapsis
To L2			Propagates until crossing the Sun-Earth-Moon L2 point's X-Z plane
First half			Propagates until the next L2 X-Z plane crossing (half a libration point orbit)
Target station-keeping 1			
Sk1	ΔV magnitude	0.0035 km/sec	Performs ΔV in the velocity direction
Second half			Propagates until the next L2 X-Z plane crossing (half a libration point orbit)
Third half			Propagates until the next L2 X-Z plane crossing (half a libration point orbit)

Table 8: Targeting profiles values used in libration point example

Profile	Independent variables			Dependent variables		
	Name	Perturbation	Max step	Name	Desired value	Tolerance
Target on B-plane	P1 ΔV	1e-4 km/sec	0.1 km/sec	B dot R	125.53 km	1e-5 km
	P2 ΔV	1e-4 km/sec	0.1 km/sec	B dot T	12554 km	1e-4 km
Target Pf maneuver	Pf ΔV	5e-8 km/sec	1e-7 km/sec	Vx in L2 frame	0 km/sec	5e-6 km/sec
Target station-keeping	Station-keeping ΔV	1e-5 km/sec	1e-4 km/sec	Vx in L2 frame	0 km/sec	1e-6 km/sec

The first target sequence starts with a launch, performs an untargeted burn to raise apogee, and then propagates through three orbits with targeted burns at the two perigee crossings. After the third perigee crossing, the sequence propagates to lunar periapsis (periselene), as the two maneuvers have boosted apogee enough to reach the Moon and the Moon has moved toward the orbit's apogee. This last segment is only executed during targeting to calculate the B-plane parameters used as dependent variables by the targeting profile. In this target sequence, the differential corrector profile targets the two maneuvers' ΔVs so that the B-plane parameters of the lunar swingby have values necessary to set the spacecraft on a course toward the L2 point. After this target sequence converges on a solution, the sequence stops at the third perigee crossing and the MCS moves to the next target sequence.

The second target sequence targets a small correction maneuver at the third perigee crossing. The sequence then propagates to lunar periapsis and continues to the libration point orbit, and then propagates through half of a libration point orbit. The target sequence's profile targets the ΔV of the correction maneuver so that at the end of this sequence the spacecraft will have no velocity in the x-direction of the L2 coordinate system, where x points from L2 to the Sun.

The third target sequence targets a station-keeping maneuver at the first plane crossing of the libration point orbit, and then propagates through an entire libration point orbit (two more plane crossings). The station-keeping ΔV is targeted so that the spacecraft has zero velocity in the L2 x direction at the end of this sequence.

Table 9 shows the number of iterations and evaluations needed by the targeting profiles in the libration point example. The table does not show a total because the profiles operate on different target sequences; thus the function evaluations $\mathbf{f}(\mathbf{x})$ are not the same for the different profiles. The table shows that Broyden's method requires more iterations than the Newton-Raphson method to solve the first and third profiles, but requires fewer evaluations to solve all three profiles.

The difference in run-time is the greatest for the first profile, where Broyden's method is 54% faster than Newton-Raphson. Since there are two independent variables in this profile, Newton-Raphson requires two more evaluations per iteration than Broyden's method requires. Because the profile takes a larger number of iterations to solve, the benefit of Broyden's method is greater than in the other profiles.

Table 9: Iterations and evaluations needed in libration point example

Profile	Newton-Raphson method		Broyden's method	
	Iterations	Evaluations	Iterations	Evaluations
Target on B-plane	9	28	10	13
Target Pf maneuver	2	5	2	4
Target station-keeping	4	9	6	8

Station-keeping

The fourth test case is a station-keeping problem for a geostationary spacecraft. The station-keeping strategy is divided into two parts: east-west station-keeping and north-south station-keeping.[9] The strategy is meant to keep the spacecraft between -59.9° and -60.1° longitude, and the inclination between 0.095° and 0.1°.

Two target sequences are used to model the station-keeping strategy, one that performs east-west station-keeping and one that performs north-south station-keeping. Table 10 describes the segments in the target sequences and Table 11 describes the targeting profiles.

Table 10: Description of segments in station-keeping example

Segment	Independent variables	Initial guess	Description
EW Station-keeping			
Coast	None		Propagates for 12 hours
EW Burn	ΔV magnitude	-0.0001 km/sec	Performs ΔV in the velocity direction
Propagate to Turnaround	None		Propagates until the longitude at the ascending node is less than the maximum longitude reached so far
NS Station-keeping			
Propagate to Node	None		Propagates to the ascending node
Coast	Duration	12 hours	Propagates for a specified duration
NS Burn	ΔV magnitude	0.007 km/sec	Performs ΔV in the orbit normal direction

Table 11: Targeting profiles values used in station-keeping example

Profile	Independent variables			Dependent variables		
	Name	Perturbation	Max step	Name	Desired value	Tolerance
EW Station-keeping	ΔV magnitude	1e-5 km/sec	1e-4 km.sec	Maximum longitude	-59.9°	0.001°
NS Station-keeping	Coast duration	1000 sec	10,000 sec	Inclination	0.1°	0.001°
	ΔV magnitude	1e-4 km/sec	0.001 km/sec	RAAN	270°	0.1°

The east-west station-keeping sequence is performed on the ascending node before the longitude is -60.09°, which is just before the west bound would be violated. The target sequence propagates to the other node and then performs a targeted maneuver. The maneuver causes the spacecraft to drift east, and the following segment propagates until gravitational perturbations cause the spacecraft to start drifting west again. The targeting profile solves for the magnitude of the maneuver so that the longitude at the turnaround point is -60.1° (the east bound). This strategy maximizes the amount of time between east-west station-keeping burns.

The north-south station-keeping sequence is performed whenever the inclination reaches 0.095°. The sequence propagates for a targeted duration after the next ascending node, and then performs a targeted ΔV in the orbit normal direction. The targeting profile solves for the duration and ΔV magnitude necessary to bring the inclination back to 0.1° while keeping the right ascension of ascending node at 270°.

The inclination boundary could be violated while the spacecraft is drifting east in the east-west station-keeping sequence. If it does, the north-south maneuver must be modeled so that the east-west maneuver gives the proper turnaround point after the north-south maneuver has affected the orbit. To account for this possibility, the Propagate to Turnaround segment in the east-west station-keeping sequence will run the north-south station-keeping sequence whenever the inclination bound is violated. If this occurs, the north-south targeting is performed during all of the evaluations of the east-west targeting, including perturbation and nominal runs of each iteration.

The station-keeping strategy is used during a 90-day propagation of the spacecraft. Over this time period, three east-west maneuvers are needed, and one north-south maneuver is needed. The north-south targeting occurs during the Propagate to Turnaround segment the second time that the east-west sequence is run. Because the station-keeping maneuvers are similar in magnitude each time that they are needed, the target sequences use the last value that they solved for as the initial guess in the next run. The initial guesses given in Table 10 are used the first time that the target sequences run. Although the north-south station-keeping sequence is run on every evaluation of the second east-west station-keeping sequence, after the first run it is always within tolerance on the initial run, so only one evaluation is required.

Table 12 shows the number of iterations and evaluations needed by the targeting profiles in the station-keeping example. The values for the north-south sequence are only given for its initial run, which is during the initial run of the second east-west sequence. For the first east-west sequence, Broyden's method requires an additional iteration but one less evaluation, so it is 14% faster than Newton-Raphson. In the second and third runs, the east-west sequence converges in one iteration, which is performed the same way for both methods. Because this targeting sequence only has a single independent variable and converges quickly the advantage of Broyden's method is less pronounced compared to the previous use cases. In the north-south sequence, Broyden's method has a greater advantage: a 43% savings in the number of evaluations. However, these evaluations are not as expensive as the east-west evaluations, which include the Propagate to Turnaround segment.

Table 12: Iterations and evaluations needed in station-keeping example

Profile	Newton-Raphson method		Broyden's method	
	Iterations	Evaluations	Iterations	Evaluations
EW Station-keeping 1	3	7	4	6
EW Station-keeping 2	1	3	1	3
NS Station-keeping	5	16	6	9
EW Station-keeping 3	1	3	1	3

Rendezvous

The fifth test case is a target sequence solving a GEO-rendezvous problem that compares the methods for targeting an orbital transfer about Earth. In this problem, a five-burn transfer is used to rendezvous from a geostationary orbit at 42,166.3 km semi-major axis, 0.0001 eccentricity, 0.1° inclination, and longitude of ascending node of 100° to a target spacecraft at the same semi-major axis and eccentricity, 1e-5° inclination, and longitude of ascending node of 260°. The rendezvous is targeted so that the spacecraft has an in-track distance of 2 km from the target spacecraft at the end of the sequence.

Table 13 describes the segments in the rendezvous target sequence and Table 14 describes the profiles used. The first two burns of the sequence are half an orbit apart and are targeted by the first profile so that after the second burn the spacecraft is in a circular orbit with a geostationary drift rate of 2° per day. The sequence then propagates until two days before it will reach the target spacecraft, and from there propagates to the line of relative nodes, which is where the spacecraft crosses the target spacecraft's orbital plane. The third burn is performed at the line of relative nodes and is targeted by the second profile so that the orbital plane becomes the same as the target spacecraft's orbital plane and the spacecraft has the same semi-major axis that it had before the burn.

Nested targeting is used in the remainder of the sequence. After the third burn, there is a Coast segment with a targeted duration, followed by two target sequences that solve for the fourth and fifth burn. In the nested targeter sequences, the fourth maneuver is targeted so that half an orbit later the spacecraft is at the same distance from the Earth as the target. The fifth burn is targeted to give the spacecraft the same semi-major axis as the target. In the main target sequence, the third profile targets the duration of the Coast segment so that the spacecraft is at the same longitude as the target at the end of the fifth burn. The final profile of the main target sequence retargets the plane change burn and targets the radial component of the fifth burn so that the spacecraft's final state is at the desired in-track location with no cross-track component and the same flight path angle as the target spacecraft. Because of the nested targeting, the semi-major axis also matches the target spacecraft, which enables the spacecraft to follow the target at the specified in-track distance.

Table 13: Description of segments in rendezvous example

Segment	Independent variables	Initial guess	Description
Initial state	None		Sets initial conditions
Change apse altitude	ΔV magnitude	0 km/sec	Performs ΔV in the velocity direction
Prop half rev	None		Propagates for half an orbit
Circularize	ΔV magnitude	0 km/sec	Performs ΔV in the velocity direction
Prop to backoff	None		Propagates to 2 days before rendezvous
Prop to rel node	None		Propagates to the line of relative nodes
Plane change maneuver	X-component of ΔV Y-component of ΔV	0 km/sec 0 km/sec	Performs specified ΔV relative to VNC axis
Prop 1 day	None		Propagates for 1 day since Prop to backoff segment
Coast	Duration	0.9 days	Propagates for specified duration
Target rendezvous			Nested target sequence
• Change apse altitude	ΔV magnitude	0 km/sec	Performs ΔV in the velocity direction
• Prop half rev	None		Propagates for half an orbit
Match Rel SMA			Nested target sequence
• Match burn	X-component of ΔV Z-component of ΔV	0 km/sec 0 km/sec	Performs specified ΔV relative to VNC axis

Table 14: Targeting profiles values used in rendezvous example

Profile	Independent variables: Name	Perturbation	Max step	Dependent variables: Name	Desired value	Tolerance
Target drift rate and eccentricity	Change apse ΔV	1e-4 km/sec	0.01 km/sec	Eccentricity	0	0.001
	Circularize ΔV	1e-4 km/sec	0.01 km/sec	Longitude drift rate	2 deg/day	1e-7 deg/sec
Plane change	Out of plane X-component	1e-4 km/sec	0.2 km/sec	Delta plane	0°	0.001°
	Out of plane Y-component	1e-4 km/sec	0.4 km/sec	Delta SMA	0 km	0.01 km
Target rel. longitude	Coast duration	1000 sec	50000 sec	Relative longitude	0°	1e-4°
Target in-track, cross-track and flight path angle	Out of plane X-component	1e-5 km/sec	1e-3 km/sec	In-track difference	2 km	0.01 km
	Out of plane Y-component	1e-5 km/sec	0.01 km/sec	Cross-track difference	0 km	0.01 km
	Match burn Z-component	1e-4 km/sec	0.1 km/sec	Relative flight path angle	0°	0.001°
Target rendezvous	Change apse ΔV	1e-4 km/sec	0.1 km/sec	Relative R-magnitude	0 km	0.001 km
Match rel SMA	Match burn X-component	1e-4 km/sec	0.1 km/sec	Relative SMA	0 km	0.001 km

Table 15 shows the number of iterations and evaluations needed by the targeting profiles in the rendezvous example. For some profiles, Broyden's method requires more iterations than the Newton-Raphson method, but in all cases Broyden's method uses fewer evaluations. Because these profiles don't use many iterations compared to some of the other test cases, the advantage of Broyden's method is not as pronounced. Still, Broyden's method saves between 14% and 33% of the runtime for these profiles. The greatest savings is for the profiles that have multiple independent variables.

Table 15: Iterations and evaluations needed in rendezvous example

Profile	Newton-Raphson method		Broyden's method	
	Iterations	Evaluations	Iterations	Evaluations
Target drift rate and eccentricity	2	7	2	5
Plane change	2	7	2	5
Target rel. longitude	3	7	4	6
Target in-track, cross-track and flight path angle	2	9	2	6
Target rendezvous	2	5	2	4
Match rel SMA	2	5	3	5

ROBUSTNESS OF METHODS

In addition to the speed advantage, Broyden's method is simpler to configure. Because Newton-Raphson uses the perturbations on every iteration, it is sensitive to the values chosen for the perturbation size. If the perturbation is too small or too big, the method may take much longer to converge on a solution or may not converge at all. In the Earth to Moon example, if the perturbations in the third profile are each increased by an order of magnitude, the method takes 11 iterations instead of 3, costing 32 evaluations. With the larger perturbations, Broyden's method, which only uses the perturbation values on the initial iteration, converges in 7 iterations instead of 3, costing 4 evaluations.

The selection of maximum step size also affects how the methods perform. If the maximum step size is chosen conservatively, more iterations are required to get the independent variables from their initial values to the solution. These additional iterations cost a smaller penalty with Broyden's method because they require fewer evaluations than they do with the Newton-Raphson method.

FUTURE WORK – PARALLELIZATION

In these examples a single processor is used for the computations. In the Newton-Raphson method, parallel processing could be used to compute the perturbation runs of an iteration simultaneously. Doing so would decrease the runtime advantage of Broyden's method shown here. If parallel processing was used the run-time cost of the Newton-Raphson method would be two evaluations per iteration, assuming that there are more processors available than the number of independent variables. For Broyden's method the first iteration would cost the run-time of two evaluations, and subsequent evaluations would still require a single evaluation. Broyden's method would still be faster for problems in which the two methods require the same number of iterations, but the advantage would not be as great when there are multiple independent variables. For problems in which Broyden's method requires more iterations than the Newton-Raphson method, Newton-Raphson could become the faster method. STK/Astrogator does not currently use parallel processing for Newton-Raphson; supporting this capability is a subject of future work.

CONCLUSION

The test cases in this paper compare using Broyden's method and the Newton-Raphson method to solve trajectory design problems with the STK/Astrogator software. In all cases both methods solve the problem within the specified tolerance. The test cases show that Broyden's method is generally faster than the Newton-Raphson method, with a speed advantage over 50% in some cases. The advantage is greater in cases in which there are multiple independent variables and the methods take more than a few iterations to converge. The advantage of Broyden's method is also greater when the perturbation is poorly chosen, or when the maximum step size is too conservative. Because Broyden's method is faster in these test cases and less sensitive to perturbation and maximum step size values, it will be the default algorithm for differential corrector profiles in version 10.0 of STK/Astrogator.

ACKNOWLEDGMENTS

The author thanks Chris Champagne and Pete Cress of Northrop Grumman Space Technology for suggesting adding Broyden's method to STK/Astrogator.

REFERENCES

[1] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, Numerical Recipes: The Art of Scientific Computing, Third Edition. Cambridge University Press, 2007. Section 9.4.

[2] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, Numerical Recipes: The Art of Scientific Computing, Third Edition. Cambridge University Press, 2007. Sections 9.6 and 18.2.

[3] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, Numerical Recipes: The Art of Scientific Computing, Third Edition. Cambridge University Press, 2007. Section 9.2.

[4] C. G. Broyden, "A Class of Methods for Solving Nonlinear Simultaneous Equations". *Mathematics of Computation* (American Mathematical Society) **19** (92): 577–593. October 1965.

[5] J. Carrico, E. Fletcher, "Software Architecture and Use of Satellite Tool Kit's Astrogator Module for Libration Point Orbit Missions", Libration Point Orbits And Applications, Parador d'Aiguablava, Girona, Spain , June 2002.

[6] M. Loucks, J. Carrico, T. Carrico, and C. Deiterich, "A Comparison of Lunar Landing Trajectory Strategies Using Numerical Simulations", International Lunar Conference. September 2005.

[7] Y. Guo, R. W. Farquhar, "New Horizons Pluto-Kuiper Belt mission: design and simulation of the Pluto-Charon encounter", *Acta Astronautica* 56: 421–429. 2005.

[8] M. Mesarch, S. Andrews, "The Maneuver Planning Process for the Microwave Anisotropy Probe (MAP) Mission," AIAA 2002-4427, AIAA/AAS Astrodynamics Specialist Conference, Monterey, CA. August 2002.

[9] C. Chao, Applied Orbit Perturbation and Maintenance. The Aerospace Press, 2005. Sections 7.1 and 7.2.