ORIGINAL ARTICLE

X.-W. Chang · Y. Guo

Huber's M-estimation in relative GPS positioning: computational aspects

Abstract When GPS signal measurements have outliers, using least squares (LS) estimation is likely to give poor position estimates. One of the typical approaches to handle this problem is to use robust estimation techniques. We study the computational issues of Huber's M-estimation applied to relative GPS positioning. First for code-based relative positioning, we use simulation results to show that Newton's method usually converges faster than the iteratively reweighted least squares (IRLS) method, which is often used in geodesy for computing robust estimates of parameters. Then for code- and carrier-phase-based relative positioning, we present a recursive modified Newton method to compute Huber's M-estimates of the positions. The structures of the model are exploited to make the method efficient, and orthogonal transformations are used to ensure numerical reliability of the method. Economical use of computer memory is also taken into account in designing the method. Simulation results show that the method is effective.

Keywords Relative GPS positioning · Huber's M-estimation · Netwon's method for minimization · Computer implementation

1 Introduction

The typical approach to GPS positioning is least squares (LS) estimation. However, LS estimation is sensitive to outliers (unspecified large errors) in the measurements; thus it will likely give poor position estimates if the measurements have outliers. There are several gross error sources which can lead to outliers in the measurement data (e.g., multipath, ionospheric delay and diffraction, interference, channel biases,

X.-W. Chang (✉) · Y. Guo
School of Computer Science, McGill University,
3480 University Street, Montreal, Quebec, Canada H3A 2A7
Tel.: +1-514-3988259,
Fax: +1-514-3983883
E-mail: chang@cs.mcgill.ca, ying.guo@mcgill.ca

satellite failure). In order to get good position estimates, outliers have to be appropriately handled.

One approach is to apply the fault detection and identification procedure (e.g., Brown 1996; Kelly 1998; Teunissen 1990). This approach uses some detection and identification techniques to detect possible unspecified errors in the input data and isolate the faulty data. Then either the faulty data are discarded or the model for position estimation is adjusted to take the unspecified errors into account. However, sometimes detection and identification by regular methods are difficult, in particular when outliers appear in multiple observations. Furthermore, in the case of limited observational data availablility, the faulty data cannot be simply discarded, otherwise, the problem of rank deficiency will arise.

Another approach is to use robust estimation techniques. The aim of robust estimation is to reduce (but not completely eliminate) the influence of outliers on the parameter estimation. It automatically identifies outliers in the observations and gives the corresponding observations less weight in the estimation. Because of the good statistical properties and the relatively low computing effort, the M-estimation techniques, especially Huber's (1964) M-estimation technique, have been widely used in robust estimation (Rieder 1996).

Robust estimation methods have been used in geodesy for many years (e.g., Fuchs 1982; Koth and Yang 1998; Xu 1989; Yang 1991, 1994; Zhu 1996; Zhong 1997; Gui and Zhang 1998; Awange and Aduol 1999 others). Recently, these techniques have been applied to GPS positioning. For example, Yang et al. (2001) proposed an adaptive robust Kalman filter for position estimation, which is some kind of combination of an adaptive Kalman filter and Huber's M-estimation. To deal with the correlated measurements, the so-called dependent equivalent weight matrix based on Huber's weight function was used in their method (see Yang 1994 for more details). Their test results show that the robust estimation can effectively resist the influence of the outliers. Wieser and Brunner (2002) present a modified Danish method for short static positioning using double-differenced carrier-phase measurements. Since the standard Danish method (Krarup et al. 1980) is not applicable to correlated observations, it was modified

to handle correlated double-differenced carrier-phase observations. Wieser and Brunner's (2002) test results indicated that their method performs significantly better than the LS method if unfavorable signal distortion occurs. Like many papers in robust M-estimation in geodesy, computer implementation of these methods was not considered in Yang et al. (2001) and Wieser and Brunner (2002), since it was not their goal.

However, computational issues are important for software design. Since computational efficiency is crucial when the amount of data is large or the application is a real-time one, fast algorithms are often needed. Numerical reliability of the algorithms has to be considered in order to avoid unnecessarily losing accuracy. One also needs to consider how to economically use computer memory in designing algorithms.

In this paper, we consider applying Huber's M-estimation technique to relative GPS positioning using the single-differenced code- and carrier-phase measurements. In the GPS literature, double-differenced measurements are often used for positioning. However, Huber's M-estimation technique cannot be applied directly to double-differenced measurements, since the noises in the measurements are correlated. Even if the correlations were not a problem, it still cannot be applied directly when the signals from the reference satellites have outliers, since all double-differenced measurements will have outliers. In fact, even if we use LS estimation, we can work with single-differenced measurements without using the double-differencing technique, see, e.g., Chang and Paige (2003) and Chang et al. (2004).

In this paper, we do not use any dynamic (or state) equations, since it is often difficult to get accurate dynamic equations in practice. However, we could extend our approach to the case where accurate dynamic equations are available. Our main goal is to address the computational issues. We will use code-based relative positioning as an example to show Newton's method is usually faster than the iteratively reweighted least squares (IRLS) method, which is widely used in geodesy for robust M-estimation (e.g., Koch 1999, Sec 3.8.2; Koth and Yang (1998); Wieser and Brunner (2002)). For code- and carrier-phase-based relative positioning, we will present a recursive modified Newton method for computing Huber's M-estimates. Due to the effect of possible outliers, we will not attempt to fix integer ambiguities. We will make full use of the structure of the positioning model to make our method efficient, and use orthogonal transformations to ensure numerical reliability. We will also consider saving computer memory in designing the method. Although our goal is not to show how significant Huber's M-estimation is to GPS positioning, we give some simulation results to demonstrate that Huber's M-estimation can give (much) better results than LS estimation when there are outliers.

The paper is organized as follows. In Sect. 2, we introduce Huber's M-estimation for a general linear model and discuss the strategies for handling a singularity problem. In Sect. 3, we introduce the mathematical models for relative positioning. In Sect. 4, we use simulation results to show Newton's method is faster than the IRLS method for Huber's M-estimation. In Sect. 5, for code- and carrier-phase-based positioning we present a recursive modified Newton method for computing Huber's M-estimates, and give some simulation results. Finally, we give a summary in Sect. 6.

Notation: Throughout this paper, we use bold lower case letters for vectors and bold upper case letters for matrices. The vector space of all real $m \times n$ matrices is denoted by $\mathcal{R}^{m \times n}$ and the vector space of real n-vectors by $\mathcal{R}^n$. The unit matrix is denoted by I or sometimes by I_n if its dimension is n by n and its ith column by e_i, while $e \equiv [1, 1, \ldots, 1]^T$ (we use $\equiv$ to mean 'is defined to be'). We use the norm $\|x\| = \sqrt{x^T x}$ for vectors. $\mathcal{E}\{\cdot\}$ denotes the expected value, and cov$\{\cdot\}$ denotes the covariance matrix, that is cov$\{x\} = \mathcal{E}\{(x - \mathcal{E}\{x\})(x - \mathcal{E}\{x\})^T\}$. $v \sim \mathcal{N}(\bar{v}, V)$ indicates that v is a normally distributed random vector with expected value $\bar{v}$ and covariance V.

2 Huber's M-estimation for a general linear model

We first describe Huber's M-estimation problem for a general linear model, and then introduce a framework for Newton's method with a line search to solve the problem (for the general Newton's method and line search strategies, see, e.g., Nocedal and Wright 1999). We discuss the advantages and disadvantages of different strategies for dealing with a singularity problem which may happen in the Newton's method and propose to use the strategy given in Chang (2005), which is suitable for solving our estimation problem for code- and carrier-phase-based relative positioning.

Suppose we have a general linear model

$$y = Ax + v, \tag{1}$$

where $y = [y_1, \ldots, y_m]^T \in \mathcal{R}^m$ is a measurement vector, $A = [a_1, \ldots, a_m]^T \in \mathcal{R}^{m \times n}$ is a design (or model) matrix with full column rank, $x \in \mathcal{R}^n$ is an unknown parameter vector to be estimated, and $v \in \mathcal{R}^m$ is a random noise vector with cov$\{v\} = \sigma^2 I$ (σ is assumed to be be known). In the ideal situation, the mean $\mathcal{E}\{v\} = 0$, but this will not hold when v includes outliers.

For a given vector $x \in \mathcal{R}^n$, we define the residual vector $r(x) \equiv y - Ax$, with ith element $r_i(x)$. Huber's M-estimate of the parameter vector is the solution of the optimization problem

$$\min_x \left\{ F(x) \equiv \sum_{i=1}^m \rho(r_i(x)) \right\}, \tag{2}$$

where ρ is a nonnegative, convex, piecewise function defined by

$$\rho(t) \equiv \begin{cases} \frac{1}{2}t^2, & |t| \leq \gamma, \\ \gamma|t| - \frac{1}{2}\gamma^2, & |t| > \gamma, \end{cases} \tag{3}$$

for some tuning constant $\gamma > 0$. Often, γ is chosen to be in the interval $[1.5\sigma, 2\sigma]$ (Koch 1999, Sec 3.8.3). The function $\rho(t)$ in the form of Eq. (3) is called the Huber function. Note that Huber's M-estimation is a mixed l_2 and l_1 minimization problem. For M-estimation, there are several other

well-known functions such as the Fair, Talwar, Tukey, and Welsh functions (Huber 1981), but the Huber function is the most popular one.

We see from Eq. (3) that $\rho(t)$ is a continuous function, with continuous and nondecreasing first derivative, since

$$\rho'(t) = \begin{cases} t, & |t| \leq \gamma, \\ \gamma \operatorname{sign}(t), & |t| > \gamma, \end{cases} \quad \rho''(t) = \begin{cases} 1, & |t| \leq \gamma, \\ 0, & |t| > \gamma. \end{cases} \quad (4)$$

Strictly speaking, $\rho(t)$ has only a right (left) second derivative at $t = -\gamma$ ($t = \gamma$), but from a practical point of view, there is no harm in defining $\rho''(\pm\gamma) = 1$, since it is very unlikely that this slight extension will have any effect on solving the minimization problem in Eq. (2).

The active index set and the inactive index set at $x \in \mathcal{R}^n$ are, respectively, defined by

$$\nu(x) \equiv \{i : |r_i(x)| \leq \gamma\}, \quad \bar{\nu}(x) \equiv \{i : |r_i(x)| > \gamma\}. \quad (5)$$

If $i \in \nu(x)$, we say that the ith equation of the model in Eq. (1) is active at x, otherwise we say it is inactive at x. The active matrix $A^\nu(x)$ of A at x is defined to be the matrix formed by the rows of A corresponding to the active equations at x. Sometimes for simplicity we just use A^ν instead of $A^\nu(x)$ if there is no confusion. We define the sign vector

$$s(x) \equiv [s_1(x), \ldots, s_m(x)]^T,$$
$$s_i(x) \equiv \begin{cases} -1, & r_i(x) < -\gamma, \\ 0, & |r_i(x)| \leq \gamma, \\ 1, & r_i(x) > \gamma, \end{cases} \quad (6)$$

and the weight matrix

$$W(x) \equiv \operatorname{diag}(w_1(x), \ldots, w_m(x)),$$
$$w_i(x) \equiv 1 - s_i^2(x) = \rho''(r_i(x)). \quad (7)$$

The objective function $F(x)$ in Eq. (2) can then be rewritten as

$$F(x) = \frac{1}{2}r(x)^T W(x) r(x)$$
$$+ \gamma s(x)^T [r(x) - \frac{1}{2}\gamma s(x)]. \quad (8)$$

Since $\frac{\partial r(x)^T}{\partial x} = -A^T$, differentiating Eq. (8) gives the gradient of $F(x)$

$$F'(x) \equiv \frac{\partial F(x)}{\partial x}$$
$$= -A^T(W(x)r(x) + \gamma s(x))$$
$$= -\sum_{i \in \nu(x)} a_i r_i(x) - \gamma \sum_{i \in \bar{\nu}(x)} a_i s_i(x). \quad (9)$$

The symmetric nonnegative definite Hessian matrix is given by

$$F''(x) \equiv \frac{\partial^2 F(x)}{\partial x \partial x^T}$$
$$= A^T W(x) A$$
$$= \sum_{i \in \nu(x)} a_i a_i^T$$
$$= A^\nu(x)^T A^\nu(x). \quad (10)$$

A general framework for Newton's method with a line search for solving Eq. (2) can be described as follows (e.g., O'Leary 1990; Madsen and Nielsen 1990; Antoch and Ekblom 1995):

Given an initial estimate x
Repeat until convergence:
  Solve the following linear system for the search direction h:

$$F''(x)h = -F'(x), \text{ or } A^\nu(x)^T A^\nu(x)h$$
$$= A^T[W(x)r(x) + \gamma s(x)], \quad (11)$$

Perform a line search and update $x := x + \hat{\alpha}h$ (see Eq. 19).

Usually the LS estimate x_{LS} for the model in Eq. (1) is taken to be the initial estimate, so that if the index set $\bar{\nu}(x_{LS})$ is empty (i.e., no outliers),

$$W(x_{LS}) = I_m,$$
$$s(x_{LS}) = 0, \quad (12)$$
$$A^T[W(x_{LS})r(x_{LS}) + \gamma s(x_{LS})] = A^T r(x_{LS}) = 0.$$

Notice that $A^T r(x_{LS}) = 0$ are the normal equations and from Eq. (11) we see $h = 0$. Thus, x_{LS} solves Eq. (2). Any h satisfying Eq. (11) with a *nonzero* $F'(x)$ is a *strict* descent direction for the functional F at x, since

$$h^T F'(x) = -h^T A^\nu(x)^T A^\nu(x) h < 0. \quad (13)$$

It can be shown (e.g., Madsen and Nielsen 1990) that there is a minimizer x of $F(x)$ such that $A^\nu(x)^T A^\nu(x)$ is nonsingular. When $A^\nu(x)^T A^\nu(x)$ is nonsingular, it is positive definite and the Cholesky factorization of $A^\nu(x)^T A^\nu(x)$ can be used to solve Eq. (11). Specifically, if $A^\nu(x)^T A^\nu(x)$ has the Cholesky factorization (e.g., Björck 1996, Sec 2.2.2)

$$A^\nu(x)^T A^\nu(x) = R^T R, \quad (14)$$

where R is upper-triangular, then h can easily be obtained by solving two triangular systems

$$R^T \tilde{h} = A^T[W(x)r(x) + \gamma s(x)], \qquad Rh = \tilde{h}. \quad (15)$$

The Cholesky factor R of $A^\nu(x)^T A^\nu(x)$ can also be obtained from the QR factorization of $A^\nu(x)$. In fact, if $A^\nu(x)$ has the QR factorization (e.g., Björck 1996, Sec 1.3)

$$A^\nu(x) = [Q, \bar{Q}]\begin{bmatrix} R \\ 0 \end{bmatrix} = QR,$$

$[Q, \bar{Q}]$ orthogonal, R upper triangular, $\quad (16)$

then we have

$$A^\nu(x)^T A^\nu(x) = R^T Q^T Q R = R^T R. \quad (17)$$

Since forming $A^\nu(x)^T A^\nu(x)$ may lose information (e.g., Björck 1996, p.44), for numerical reliability, we prefer to find R by the QR factorization in Eq. (16).

The main computational cost in each iteration is then that of computing the QR factorization. Since the active matrices for two consecutive iterates usually differ by just a few rows, updating/downdating techniques for the QR factorization (e.g., Björck 1996, Sec 3.2) can be used to compute the QR factorization of $A^\nu(x)$ during the iterations for efficiency.

If during the iterative process there are more than $m - n$ large residuals beyond the tuning constant γ, then the number of rows of $A^\nu(x)$ is smaller than the number of its columns, so $A^\nu(x)^T A^\nu(x)$ is singular. There are several strategies to handle this problem. For example, in Antoch and Ekblom (1995), if $A^\nu(x)^T A^\nu(x)$ is found to be singular, then it is replaced by $A^\nu(x)^T A^\nu(x) + \epsilon I$. The shortcoming with this approach is that updating/downdating of the matrix factorization is computationally expensive when the matrix becomes a singular matrix from a nonsingular matrix, or vice versa, since it is computationally expensive to go from the factorization of $A^\nu(x)^T A^\nu(x)$ to $A^\nu(x)^T A^\nu(x) + \epsilon I$, or vice versa. In O'Leary (1990), the strategy is to use a very large tuning constant at the beginning, then gradually decrease its value to the desired value over the first four steps of the iteration. However, no implementation details for this strategy were given in O'Leary (1990), nor was a guarantee given that this strategy would always work.

In this paper, we use the following strategy (Chang 2005) to handle the singularity problem. If the active matrix $A^\nu(x)$ is not of full column rank, we choose a row vector from those a_i^T with $i \in \bar{\nu}(x)$ according to some criterion and add this row vector to $A^\nu(x)$. We continue this process until the updated $A^\nu(x)$ has full column rank. For simplicity, the updated $A^\nu(x)$ is still called an active matrix. For a general matrix A such as that in GPS code-based relative positioning (see Sect. 4), a sensible criterion is that the chosen row vector corresponds to the smallest residual in magnitude. In practice, we may know when the updated $A^\nu(x)$ has full column rank. For GPS code-based relative positioning, when the updated $A^\nu(x)$ becomes square, we can reasonably assume that it is nonsingular. For GPS code- and carrier-phase-based relative positioning, the matrix A has some special structures (see Sect. 5.1), and we will show how to choose row vectors to make the updated $A^\nu(x)$ have full column rank. After a full column rank $A^\nu(x)$ is found, we solve Eq. (11) for the search direction h. Note that although now $A^\nu(x)$ may not be the true active matrix at x any more, we still have Eq. (13). Thus h is still a descent direction and the optimization problem Eq. (2) remains solved. Since the singularity problem does not occur often (unless the tuning constant γ in Eq. (3) is too small or there are many outliers), this strategy is unlikely to slow down the convergence.

The optimal step length $\hat{\alpha}$ in the line search can be found exactly. Write

$$\phi(\alpha) \equiv F(x + \alpha h) = \sum_{i=1}^{m} \rho(y_i - a_i^T(x + \alpha h)). \qquad (18)$$

Since $\phi(\alpha)$ is the sum of nonnegative, convex, piecewise defined functions of α, with each piece being either quadratic or linear (see Eqs. 2–4), it too is a nonnegative, convex,

piecewise defined function with each piece being quadratic (possibly linear). Therefore, $\phi'(\alpha)$ must be piecewise, with each piece linear (possibly constant), and we can find exactly a minimizer $\hat{\alpha}$ of $\phi(\alpha)$, i.e.,

$$\hat{\alpha} = \arg\min_{\alpha} \phi(\alpha). \qquad (19)$$

This $\hat{\alpha}$ is a zero of $\phi'(\alpha)$. An efficient method for computing $\hat{\alpha}$ can be found in Madsen and Nielsen (1990). Its computational cost is usually negligible compared with that of solving the linear system in Eq. (11).

The commonly used method for solving the robust M-estimation problem in geodesy is the IRLS method (e.g., Koch 1999). Note that Eq. (11) can also be written

$$A^T W(x) A h = A^T [W(x) r(x) + \gamma s(x)]. \qquad (20)$$

If we replace $W(x)$ on the left-hand side of Eq. (20) by $D(x) = \text{diag}(d_1, \ldots, d_m)$ with $d_i = 1$ if $|r_i(x)| \leq \gamma$ or $d_i = \gamma/|r_i(x)|$ if $|r_i(x)| > \gamma$, then we can easily verify that Eq. (20) can be written

$$A^T D(x) A x_{\text{new}} = A^T D(x) y, \qquad (21)$$

where $x_{\text{new}} = x + h$. The iterative method in which the iteration sequence defined by Eq. (21) (i.e., x_{new} is a new iterate) is just the IRLS method for Huber's M-estimation. Choosing different $D(x)$ in Eq. (21) will lead to other robust M-estimation methods, such as the Danish method.

Since it is easy to understand and implement the IRLS method, this is probably why it is often used in geodesy. Unlike Newton's method, however, we cannot use the updating/downdating techniques for the QR factorization to compute the factorization in Eq. (14) during the iterations, though we can use the updating/downdating techniques for the Cholesky factorization to do it (Wolke 1992), which may lose accuracy. Furthermore, usually the IRLS method has a linear convergence rate, while Newton's method has quadratic convergence rate (Osborne 1985, Sec 5.4 and Sec 5.6; Antoch and Ekblom 1995). Since the cost of the line search is usually negligible compared with that of solving the linear systems during the iterations, Newton's method is usually more computationally efficient than the IRLS method.

3 Mathematical models

We consider relative positioning based on L1 single-frequency code and carrier-phase measurements. Ideally when the distance between the stationary receiver and roving receivers is short (say, shorter than 10 km), the signals received by the two receivers from the same satellite have almost the same ionospheric refraction and tropospheric refraction. The single difference code and carrier-phase measurement equations for satellite i at epoch k can be written as (cf. Chang and Paige 2003)

$$\rho_k^i = (e_k^i)^T x_k + c\delta t_k + \mu_k^i, \qquad (22)$$

$$\phi_k^i = (e_k^i)^T x_k + c\delta t_k + \lambda N^i + \nu_k^i, \qquad (23)$$

where all terms are in units of meters, ρ_k^i is the single-differenced code measurement, ϕ_k^i is the single-differenced carrier-phase measurement, x_k is the baseline vector pointing from the stationary receiver to the roving receiver, e_k^i is the unit vector pointing from the midpoint of the baseline to satellite i, c is the speed of light, δt_k is the single-differenced receiver clock error, λ is the wavelength of L1 carrier ($\lambda \approx$ 19 cm), N^i is the single-differenced ambiguity including single-differenced initial phase of the receiver-generated signals and so is not an integer, μ_k^i is single-differenced noise including multipath for the code measurement, and v_k^i is single-differenced noise including multipath for the carrier-phase measurement.

Suppose there are m visible satellites. Define

$$y_k^\rho = \begin{bmatrix} \rho_k^1 \\ \vdots \\ \rho_k^m \end{bmatrix}, \quad y_k^\phi = \begin{bmatrix} \phi_k^1 \\ \vdots \\ \phi_k^m \end{bmatrix}, \quad E_k = \begin{bmatrix} (e_k^1)^{\mathrm{T}} \\ \vdots \\ (e_k^m)^{\mathrm{T}} \end{bmatrix}. \ \beta_k = c\delta t_k,$$

$$a = \lambda \begin{bmatrix} N^1 \\ \vdots \\ N^m \end{bmatrix}, \quad v_k^\rho = \begin{bmatrix} \mu_k^1 \\ \vdots \\ \mu_k^m \end{bmatrix}, \quad v_k^\phi = \begin{bmatrix} v_k^1 \\ \vdots \\ v_k^m \end{bmatrix}. \quad (24)$$

Then from Eqs. (22) and (23) we have

$$y_k^\rho = E_k x_k + \beta_k e + v_k^\rho, \quad (25)$$

$$y_k^\phi = E_k x_k + \beta_k e + a + v_k^\phi, \quad (26)$$

where, as in some GPS literature (e.g., Tiberius 1998, Sec 3.4), we assume that in the ideal situation,

$$v_k^\rho \sim \mathcal{N}(0, \sigma_\rho^2 I_m), \qquad v_k^\phi \sim \mathcal{N}(0, \sigma_\phi^2 I_m), \quad (27)$$

and $v_k^\rho, v_l^\rho, v_k^\phi$, and v_l^ϕ are uncorrelated to each other for any epochs k and l ($k \neq l$). Here we give a remark about these assumptions. In practice, the noise variances depend on the elevation angles of the satellites (e.g., Tiberius and Kenselaar 2003), but for either the code measurement equations in Eq. (25) or the carrier-phase measurement equations in Eq. (26), we could simply scale them so that they have the same variance. In practice, the measurements may be correlated between channels, between measurement types and between epochs. However, an accurate characterization of the correlation is very difficult, although there have been some attempts in the literature (e.g., Tiberius and Kenselaar 2003).

Notice that in E_k, e_k^i depends on the baseline vector x_k. Once x_k is known, E_k will be known. In other words, Eqs. (25) and (26) are nonlinear. However, E_k is not sensitive to changes in x_k, since the distance from a satellite to any of the two receivers is far larger than the baseline. Therefore, we can use an approximation to x_k, say, our estimate of x_{k-1} at epoch $k - 1$, to compute the approximation to E_k. Usually this approximation is good enough, but after obtaining an estimate of x_k at epoch k, we can recompute E_k, and do more iterations if necessary. For the first epoch, we set the initial estimate of x_1 to be a zero vector and then compute the corresponding E_1 (i.e., we take e_1^i to be the unit vector

pointing from the stationary receiver to satellite i). Since E_1 computed by this initial x_1 is not very accurate, it is recomputed one more time when the new estimate of x_1 is obtained from our computational method. Therefore, from now on, we just assume all E_k are known.

4 Computation for code based relative positioning

In this section, we apply Newton's method directly to code-based relative positioning. The purpose is to use simulation results to show that for computing Huber's M-estimates Newton's method converges faster than the IRLS method, which is often used in geodesy, and demonstrate that Huber's estimation gives better position estimates than the LS estimation when there are outliers in the measurements.

All our computations were performed in MATLAB 6.5 on a Celeron PC running Windows XP. The GPS satellite almanac data in YUMA format for the week of June 30th to July 6th 1998 were used in the simulations. The roving receiver was assumed to be on board an aircraft circling horizontally with center directly above the reference station at a constant speed of 100 m/s. The baseline was 1 km. For each epoch, a set of code measurement data from eight visible satellites were used. The data sampling rate was 1 Hz. At each epoch, the single-differenced code measurements were simply constructed by

$$y_k^\rho = \bar{y}_k + c\delta t_k e + v_k^\rho + b^\rho, \quad (28)$$

where the ith component of $\bar{y}_k$ is the difference of the true range between satellite i and the stationary receiver and the true range between satellite i and the roving receiver, δt_k is the difference between the clock offsets of the stationary receiver and the roving receiver, both of which were modeled by white noise input to a second-order Markov process based on Axelrad and Brown (1996, pp 417–418), $v_k^\rho \sim \mathcal{N}(0, \sigma_\rho^2 I)$ with $\sigma_\rho = 1$ m (for typical values of the standard deviations of code and carrier-phase noises, see Misra and Enge 2001, p153), and b^ρ is the outlier vector.

We used the IRLS method and Newton's method to compute Huber's M-estimates of the positions. For consistency, both methods used the same stopping criterion: the iteration process stops when the difference between the position estimates at two consecutive iteration steps is less than 0.1 m. Thus the position estimates computed by the two methods have the same numerical accuracy. The tuning constant γ was set to be 1.5 (Koch 1999, Sec 3.8.3). We took the outlier vector $b^\rho = [0, 12, 0, 0, 8, 0, 0, 0]^{\mathrm{T}}$ (in meters) for each epoch. In order to reduce the random effect on the results, we performed 100 simulation runs for the same satellite geometries (i.e., the geometries for different runs at the same epoch are the same), and each run had 200 epochs.

Figure 1 displays the *average* number of iterations for Newton's method and the IRLS method at each epoch. From Fig. 1, we see that Newton's method converges faster than the IRLS method, as the literature showed. On average, the former takes only about three or four iterations, while the
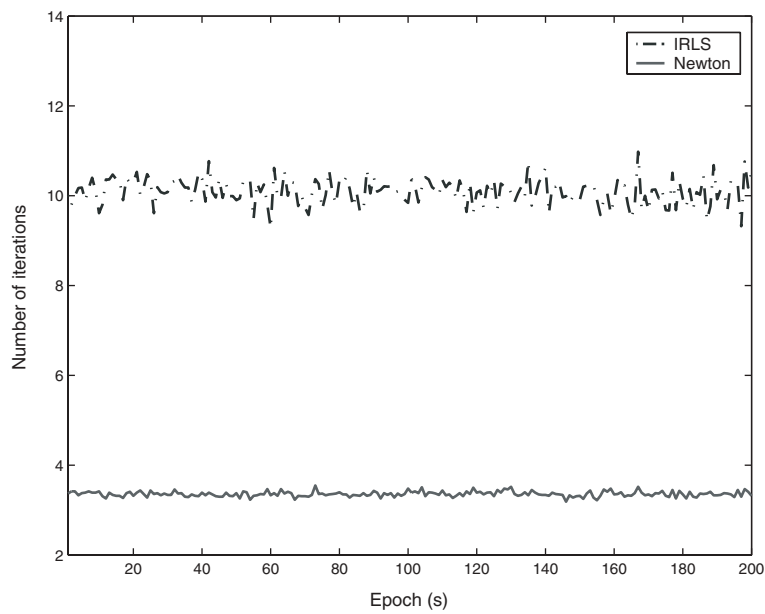
Fig. 1 Average number of iterations for the two methods using simulated code data
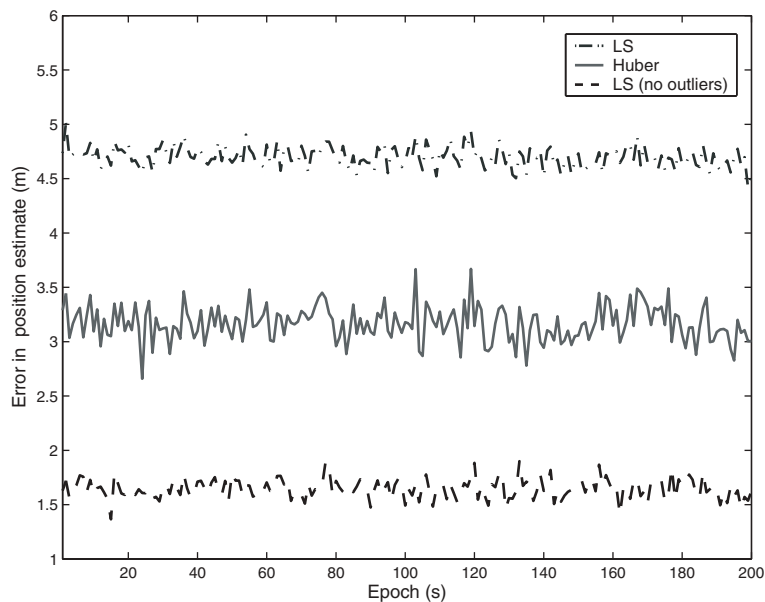

Fig. 2 Average errors in the position estimates using simulated code data

latter takes about ten iterations. Since in each iteration the computational costs of the two methods are more or less the same if both use updating/downdating techniques, these simulation results indicate that Newton's method is more computationally efficient than the IRLS method.

Figure 2 shows the *average* position error in Huber's M-estimate at each epoch, which were computed by Newton's method, and the *average* position errors in the LS estimates which were computed using the QR factorization method. For comparison, Fig. 2 also gives the average position errors in the LS estimates when the measurements do not have outliers. We observe that Huber's M-estimation can

reduce the effect of outliers – giving better position estimates than the LS estimation, although the estimates are not as good as the LS estimates without outliers, indicating that Huber's M-estimatation cannot completely eliminate the effect of outliers.

5 Computation for code- and carrier-phase-based positioning

In this section, we first present a recursive modified Newton method for computing Huber's M-estimates based on

single-differenced code and carrier-phase measurements, then give some simulation results to show that our method is effective.

5.1 Numerical method

In order to apply Huber's M-estimation, we have to scale the code or carrier-phase measurement equations so that they have the same variances. Multiplying Eq. (25) by the ratio $\sigma_r \equiv \sigma_\phi / \sigma_\rho$, and then combining it with Eq. (26), we obtain

$$
\begin{bmatrix} y_k^\phi \\ \sigma_r y_k^\rho \end{bmatrix} = \begin{bmatrix} e & E_k \\ \sigma_r e & \sigma_r E_k \end{bmatrix} \begin{bmatrix} \beta_k \\ x_k \end{bmatrix} + \begin{bmatrix} I \\ 0 \end{bmatrix} a + \begin{bmatrix} v_k^\phi \\ \sigma_r v_k^\rho \end{bmatrix},
$$
$$
\begin{bmatrix} v_k^\phi \\ \sigma_r v_k^\rho \end{bmatrix} \sim \mathcal{N}(0, \sigma_\phi^2 I_{2m}). \tag{29}
$$

Define

$$
y_k \equiv \begin{bmatrix} y_k^\phi \\ \sigma_r y_k^\rho \end{bmatrix} \in \mathcal{R}^{2m}, \quad B_k \equiv \begin{bmatrix} e & E_k \\ \sigma_r e & \sigma_r E_k \end{bmatrix} \in \mathcal{R}^{2m \times 4}, \tag{30}
$$
$$
z_k \equiv \begin{bmatrix} \beta_k \\ x_k \end{bmatrix} \in \mathcal{R}^{m+4}, \quad C \equiv \begin{bmatrix} I \\ 0 \end{bmatrix} \in \mathcal{R}^{2m \times m},
$$
$$
v_k \equiv \begin{bmatrix} v_k^\phi \\ \sigma_r v_k^\rho \end{bmatrix} \in \mathcal{R}^{2m}. \tag{31}
$$

Then Eq. (29) can be rewritten as

$$
y_k = B_k z_k + C a + v_k, \qquad v_k \sim \mathcal{N}(0, \sigma_\phi^2 I_{2m}). \tag{32}
$$

Combining the above equations for $k = 1, 2, \ldots$ gives

$$
\begin{bmatrix} y_1 \\ \vdots \\ y_{k-1} \\ y_k \end{bmatrix} = \begin{bmatrix} B_1 & & & C \\ & \ddots & & \vdots \\ & & B_{k-1} & C \\ & & & B_k & C \end{bmatrix} \begin{bmatrix} z_1 \\ \vdots \\ z_{k-1} \\ z_k \\ \hline a \end{bmatrix} + \begin{bmatrix} v_1 \\ \vdots \\ v_{k-1} \\ v_k \end{bmatrix},
$$
$$
k = 1, 2, \ldots, \tag{33}
$$

or equivalently (with obvious notation)

$$
y_{[k]} = [B_{[k]}, C_{[k]}] \begin{bmatrix} z_{[k]} \\ a \end{bmatrix} + v_{[k]} \equiv A_{[k]} x_{[k]} + v_{[k]},
$$
$$
k = 1, 2, \ldots. \tag{34}
$$

Note that each matrix $A_{[k]}$ has a block angular structure. Suppose there are at least four visible satellites at any epoch. It is reasonable to assume that a matrix formed by any four rows of $[e, E_j]$ for any j is nonsingular (this assumption will be used later). Of course $[e, E_j]$ will have full column rank. Then it is straightforward to show that $A_{[k]}$ has full column rank.

We will apply Huber's M-estimation to Eq. (33) or (34) at epoch k. In other words, all measurement equations from epoch 1 to epoch k will be used for the estimation at epoch k. We will obtain not only the estimate for the position at epoch k, which is called the *filtered estimate*, but also the estimates for positions at all previous epochs, which are called *smoothed estimates*. For the same position, its smoothed

estimates obtained at later epochs are usually more accurate than its filtered estimate, since the former use more information than the latter.

At an estimate of $x_{[k]}$, we can find the corresponding active matrix of $A_{[k]}$. For convenience, we also call $[B_j^v, C^v]$ which is formed by the rows of $[B_j, C]$ corresponding to the active equations the active matrix of $[B_j, C]$ at the estimate, and call B_j^v the active matrix of B_j at the estimate. For each epoch k, we could directly apply Newton's method to Eq. (33) or (34) to compute Huber's M-estimate of $x_{[k]}$, but this is not computationally efficient. At epoch k, when we compute Huber's M-estimate of $x_{[k]}$, we should use the results available at epoch $k-1$. Also the structures of $A_{[k]}$ should be exploited to make the algorithm efficient. In Chang (2005), a recursive modified Newton method is proposed to compute Huber's M-estimates for a model more general than Eq. (33), which may arise from other applications such as photogrammetry, geodetic survey and meteorology. We would like to apply the techniques presented in Chang (2005) to the model in Eq. (33), where B_j and C have some special structures – the bottom $m \times 4$ submatrix of B_j is a multiple of its top $m \times 4$ submatrix; and the top and bottom block of C are an identity matrix and a zero matrix, respectively.

From the general framework for Newton's method, we see that at epoch k the main cost of computing Huber's M-estimate is the cost of finding the R-factor of the QR factorization of the active matrix $A_{[k]}^v$ (see Eq. 16) in each iteration step. As we mentioned in Sect. 2, for computational efficiency, updating/downdating techniques of the QR factorization of the active matrix $A_{[k]}^v$ have to be used. During the iterations at epoch k, a measurement equation at any previous epoch may become an active equation from an inactive equation or vice versa. This makes updating/downdating of the QR factorization of the active matrices very complicated. Furthermore the Q-factors of all $[B_j, C]$ have to be stored for updating/downdating use at later epochs. When k is large, this may cause a computer memory problem. Thus we would like to modify Newton's method in the following way.

When we compute Huber's M-estimate of $x_{[k]}$ at epoch k, we do not update the active matrix of $[B_j, C]$ at Huber's M-estimate of $x_{[j]}$ obtained at epoch j for $j = 1, \ldots, k-1$, and only update the active matrix of $[B_k, C]$ at each iterate. In other words, in each iteration step at epoch k we will use a modified active matrix (whose first $k-1$ row blocks will not be changed during the iterations) to replace the true active matrix $A_{[k]}^v$ at the iterate. Then updating/downdating of the QR factorization of the modified active matrix will be much simpler than that of the actual active matrix and may involve much fewer floating point operations. The details of the computation will be given in this section. With the above modification, we can easily show that we will still obtain a descent direction at each iterate (unless the convergence has been reached), since we actually use a new symmetric positive definite matrix to replace the left-hand side of Eq. (11) without changing the definition of the right-hand side of Eq. (11). Since the descent direction is sub-optimal at each iterate, it may take more iterations than the standard Newton's method.

However, the numerical tests given in Sect. 5.2 indicate that this is not a concern.

For later use, here we describe the computation of the QR factorization of $[B_k, C]$. Notice that $[B_k, C]$ has a special structure, which should be used in the computation for efficiency. Define the orthogonal matrix

$$G \equiv \begin{bmatrix} cI & -sI \\ sI & cI \end{bmatrix},$$

$$\alpha \equiv \sqrt{1 + \sigma_r^2}, \quad c \equiv 1/\alpha, \quad s \equiv \sigma_r/\alpha. \tag{35}$$

Then multiplying $[B_k, C]$ by G^{T} from the left gives

$$
G^{\mathrm{T}}\left[B_k | C\right] = \begin{bmatrix} cI & sI \\ -sI & cI \end{bmatrix} \begin{bmatrix} e & E_k \mid I \\ \sigma_r e & \sigma_r E_k \mid 0 \end{bmatrix}
$$

$$
= \begin{bmatrix} \alpha e & \alpha E_k \mid cI \\ 0 & 0 \mid -sI \end{bmatrix}. \tag{36}
$$

Use the Householder transformations (e.g., Björck 1996, Sec 2.3) to compute the QR factorization:

$$
H_k^{\mathrm{T}} \begin{bmatrix} \alpha e & \alpha E_k \mid cI \\ 0 & 0 \mid -sI \end{bmatrix} = \begin{bmatrix} R_k \mid \hat{R}_k \\ 0 \mid \bar{R}_k \end{bmatrix}, \tag{37}
$$

where $H_k \in \mathcal{R}^{2m \times 2m}$ is the product of $m + 4$ Householder transformations, R_k is a 4×4 nonsingular upper triangular matrix, and $\bar{R}_k$ is a $(2m - 4) \times m$ upper triangular matrix. Here the Householder transformations can be implemented to take advantage of the the structure of the matrix. Combining Eqs. (36) and (37) gives the QR factorization of $[B_k, C]$:

$$
\begin{bmatrix} Q_k^{\mathrm{T}} \\ \bar{Q}_k^{\mathrm{T}} \end{bmatrix} \left[B_k | C\right] = \begin{bmatrix} R_k \mid \hat{R}_k \\ 0 \mid \bar{R}_k \end{bmatrix}, \quad [Q_k, \bar{Q}_k] \equiv G H_k, \tag{38}
$$

where $[Q_k, \bar{Q}_k]$ is orthogonal, and Q_k^{T} and $[R_k, \hat{R}_k]$ have the same number of rows. Here the Q-factor $[Q_k, \bar{Q}_k]$ needs to be formed and stored. It is used during the iteration process at epoch k. After epoch k, it will not be used any more and can be discarded.

In the following, we will discuss the computation at epoch k. First, we consider the initial case $k = 1$. Note that the model is

$$
\begin{bmatrix} y_1^\phi \\ \sigma_r y_1^\rho \end{bmatrix} = \begin{bmatrix} e & E_1 & I \\ \sigma_r e & \sigma_r E_1 & 0 \end{bmatrix} \begin{bmatrix} \beta_1 \\ x_1 \\ a \end{bmatrix} + \begin{bmatrix} v_1^\phi \\ \sigma_r v_1^\rho \end{bmatrix}, \tag{39}
$$

or

$$
y_1 = [B_1, C] \begin{bmatrix} z_1 \\ a \end{bmatrix} + v_{[1]} = A_{[1]} x_{[1]} + v_{[1]}. \tag{40}
$$

We can directly apply Newton's method given in Sect. 2 to the model in Eq. (40) to compute Huber's M-estimate, with the initial estimate taken to be the LS estimate, which can be computed by the QR factorization of $[B_1, C]$ (cf. Eq. 38). From the structure of the matrix C, we observe that Huber's M-estimation cannot reduce the influence of outliers in carrier-phase measurements, since outliers will be absorbed into the unknown ambiguities, i.e., we cannot distinguish outliers

and ambiguities (this is also true for $k > 1$ if all carrier-phase measurement equations at different epochs have the same outlier vector). Therefore, the residuals for the carrier-phase measurement equations at an iteration are likely to be small, and all the carrier-phase measurement equations are likely to be active.

Also, in theory, there is Huber's estimate at which the active matrix of $A_{[1]}$ has full column rank (see the statement after Eq. 13), so all carrier-phase measurement equations must be active at the estimate, otherwise from the structure of C we observe that the active matrix will have at least one zero column. In our implementation, we force the active matrix at any iteration to include all the m row vectors corresponding to the m carrier-phase measurement equations. If the number of rows of an active matrix is less than $m + 4$ (the number of its columns), then the active matrix does not have full column rank and we use the strategy proposed in Sect. 2 to handle this singularity problem. Specifically, we add to the active matrix the row vectors corresponding to the code measurement equations which are not active and have the smallest residuals such that the modified active matrix becomes square, then it is easy to show it is nonsingular under our earlier assumption that any four rows of $[e, E_j]$ is nonsingular.

Now we consider the general case $k > 1$. Let Huber's M-estimate of $x_{[j]}$ at any epoch j be denoted by $x_{[j|j]} = [z_{1|j}^{\mathrm{T}}, \dots, z_{j|j}^{\mathrm{T}}, a_j^{\mathrm{T}}]^{\mathrm{T}}$ and let the active matrix of $[B_j, C]$ at $x_{[j|j]}$ by $[B_{j|j}, C_j]$. At the end of epoch $k - 1$, we obtain

$$A_{[k-1|k-1]} \equiv [B_{[k-1|k-1]}, C_{[k-1|k-1]}]$$

$$
\equiv \begin{bmatrix} B_{1|1} & & C_1 \\ & \ddots & \vdots \\ & & B_{k-1|k-1} & C_{k-1} \end{bmatrix}, \tag{41}
$$

where we assume that $B_{j|j}$ (for $j = 1, \dots, k - 1$) has full column rank (we will discuss how to ensure this later), so $A_{[k-1|k-1]}$ has full column rank. For computational efficiency, we will use $A_{[k-1|k-1]}$ in lieu of true active matrices of $A_{[k-1]}$ at later iterates for computing descent directions, and will not update $A_{[k-1|k-1]}$ any more.

Suppose at the end of epoch $k - 1$, we have obtained the QR factorization of $A_{[k-1|k-1]}$:

$$
\begin{bmatrix} Q_{[k-1|k-1]}^{\mathrm{T}} \\ \bar{Q}_{[k-1|k-1]}^{\mathrm{T}} \end{bmatrix} A_{[k-1|k-1]}
$$

$$
= \begin{bmatrix} R_{1|1} & & \hat{R}_{1|1} \\ & \ddots & \vdots \\ & & R_{k-1|k-1} & \hat{R}_{k-1|k-1} \\ \hline & & & \tilde{R}_{[k-1]} \\ & & & 0 \end{bmatrix}, \tag{42}
$$

where $[Q_{[k-1|k-1]}, \bar{Q}_{[k-1|k-1]}]$ is orthogonal, and $R_{j|j}$ for $j = 1, \dots, k - 1$ and $\tilde{R}_{[k-1]}$ are nonsingular upper triangular. Here the Q-factor $[Q_{[k-1|k-1]}, \bar{Q}_{[k-1|k-1]}]$ is not used in computation and so is not formed and stored.

At epoch k, let the initial estimate for $x_{[k]}$ be denoted by

$$x_{[k|k]}^{(0)} \equiv [(z_{1|k}^{(0)})^{\mathrm{T}}, \dots, (z_{k-1|k}^{(0)})^{\mathrm{T}}, (z_{k|k}^{(0)})^{\mathrm{T}}, (a_k^{(0)})^{\mathrm{T}}]^{\mathrm{T}}. \tag{43}$$

Naturally we take

$$z_{j|k}^{(0)} = z_{j|k-1}, \quad j = 1, \ldots, k-1, \qquad a_k^{(0)} = a_{k-1}. \quad (44)$$

For the initial estimate of z_k, noting that the measurement equations at epoch k are $y_k = B_k z_k + C a + v_k$, we take

$$z_{k|k}^{(0)} = \arg\min_{z_k} \|(y_k - C a_{k-1}) - B_k z_k\|. \quad (45)$$

Using the QR factorization in Eq. (38), we can easily find $z_{k|k}^{(0)}$ in Eq. (45) by solving the upper triangular system

$$R_k z_{k|k}^{(0)} = Q_k^{\mathrm{T}}(y_k - C a_{k-1}). \quad (46)$$

After finding $z_{k|k}^{(0)}$, we can obtain the modified active matrix of $A_{[k]}$ at $x_{[k|k]}^{(0)}$:

$$A_{[k|k]}^{(0)} \equiv \begin{bmatrix} B_{1|1} & & & C_1 \\ & \ddots & & \vdots \\ & & B_{k-1|k-1} & C_{k-1} \\ & & & B_{k|k}^{(0)} \ C_k^{(0)} \end{bmatrix}, \quad (47)$$

where $[B_{k|k}^{(0)}, C_k^{(0)}]$ is the active matrix of $[B_k, C_k]$ at $x_{[k|k]}^{(0)}$. If $[B_{k|k}^{(0)}, C_k^{(0)}]$ corresponds to measurements from less than four different satellites, then from the structure of B_k (see Eq. 30) we can conclude that $B_{k|k}^{(0)}$ does not have full column rank, so $A_{[k|k]}^{(0)}$ does not have full column rank. In order to handle this problem, we use the strategy proposed in Sect. 2.

Specifically, we add to the active matrix $[B_{k|k}^{(0)}, C_k^{(0)}]$ the row vectors from the rest of $[B_k, C]$ corresponding to other satellites and smallest residuals so that the row vectors of the updated $[B_{k|k}^{(0)}, C_k^{(0)}]$ corresponds to four different satellites. Then we can easily show that the updated $B_{k|k}^{(0)}$ has full column rank under the earlier assumption that any four rows of $[e, E_j]$ for any j is nonsingular. In our later iterations at epoch k, we always use this strategy to ensure that the active matrices of B_k have full column rank. We then use $A_{[k|k]}^{(0)}$ in lieu of the true active matrix of $A_{[k]}$ at $x_{[k|k]}^{(0)}$ for computing the search direction $h_{[k]}^{(0)}$ at $x_{[k|k]}^{(0)}$, i.e., we would like to solve the following linear system for $h_{[k]}^{(0)}$ (cf. Eq. 11):

$$\left(A_{[k|k]}^{(0)}\right)^{\mathrm{T}} A_{[k|k]}^{(0)} h_{[k]}^{(0)} = \left(A_{[k|k]}^{(0)}\right)^{\mathrm{T}}\left[W(x_{[k|k]}^{(0)}) \cdot r(x_{[k|k]}^{(0)}) + \alpha s(x_{[k|k]}^{(0)})\right]. \quad (48)$$

In order to solve Eq. (48), we seek the QR factorization of $A_{[k|k]}^{(0)}$. Applying the QR downdating technique to Eq. (38), we can obtain the QR factorization of $[B_{k|k}^{(0)}, C_k^{(0)}]$:

$$\begin{bmatrix} (Q_{k|k}^{(0)})^{\mathrm{T}} \\ \bar{Q}_{k|k}^{(0)\mathrm{T}} \end{bmatrix} \begin{bmatrix} B_{k|k}^{(0)} \big| C_k^{(0)} \end{bmatrix} = \begin{bmatrix} R_{k|k}^{(0)} \big| \hat{R}_{k|k}^{(0)} \\ \hline 0 \big| \bar{R}_{k|k}^{(0)} \end{bmatrix}. \quad (49)$$

Then using the QR factorizations (42) and (49), we obtain from Eq. (47) that

$$\begin{bmatrix} Q_{[k-1|k-1]}^{\mathrm{T}} & \\ & (Q_{k|k}^{(0)})^{\mathrm{T}} \\ \hline \bar{Q}_{[k-1|k-1]}^{\mathrm{T}} & \\ & (\bar{Q}_{k|k}^{(0)})^{\mathrm{T}} \end{bmatrix} A_{[k|k]}^{(0)}$$
$$= \begin{bmatrix} R_{1|1} & & & \hat{R}_{1|1} \\ & \ddots & & \vdots \\ & & R_{k-1|k-1} & \hat{R}_{k-1|k-1} \\ & & R_{k|k}^{(0)} & \hat{R}_{k|k}^{(0)} \\ \hline & & & \tilde{R}_{[k-1]} \\ & & & 0 \\ & & & \bar{R}_{k|k}^{(0)} \end{bmatrix}. \quad (50)$$

The next step is to compute the following QR factorization by Householder transformations:

$$(\tilde{Q}_{[k]}^{(0)})^{\mathrm{T}} \begin{bmatrix} \tilde{R}_{[k-1]} \\ \bar{R}_{k|k}^{(0)} \end{bmatrix} = \begin{bmatrix} \tilde{R}_{[k]}^{(0)} \\ 0 \end{bmatrix}. \quad (51)$$

In this implementation, the special structures of $\tilde{R}_{[k-1]}$ and $\bar{R}_{k|k}^{(0)}$ are used for computational efficiency. Here again the Q-factor $\tilde{Q}_{[k]}$ does not need to be formed and stored. From Eqs. (50) and (51), we see that there exists an orthogonal matrix $[Q_{[k|k]}^{(0)}, \bar{Q}_{[k|k]}^{(0)}]$ such that

$$\begin{bmatrix} (Q_{[k|k]}^{(0)})^{\mathrm{T}} \\ (\bar{Q}_{[k|k]}^{(0)})^{\mathrm{T}} \end{bmatrix} A_{[k|k]}^{(0)}$$
$$= \begin{bmatrix} R_{1|1} & & & \hat{R}_{1|1} \\ & \ddots & & \vdots \\ & & R_{k-1|k-1} & \hat{R}_{k-1|k-1} \\ & & R_{k|k}^{(0)} & \hat{R}_{k|k}^{(0)} \\ \hline & & & \tilde{R}_{[k]}^{(0)} \\ & & & 0 \end{bmatrix}. \quad (52)$$

This is the QR factorization of $A_{[k|k]}^{(0)}$. Here we do not form or store the Q-factor. Solving Eq. (48) by using the QR factorization in Eq. (52), we obtain the search direction $h_{[k]}^{(0)}$. Then applying the line search technique, we get the next iterate $x_{[k|k]}^{(1)}$.

Now we can continue the iteration process. After obtaining $x_{[k|k]}^{(1)}$, we can determine the active matrix $[B_{k|k}^{(1)}, C_k^{(1)}]$ of $[B_k, C]$ at $x_{[k|k]}^{(1)}$. The modified active matrix $A_{[k|k]}^{(1)}$ of $A_{[k]}$ at $x_{[k|k]}^{(1)}$ is then obtained by replacing $[B_{k|k}^{(0)}, C_k^{(0)}]$ at the bottom of $A_{[k|k]}^{(0)}$ in Eq. (47) with $[B_{k|k}^{(1)}, C_k^{(1)}]$. Then we just repeat the process in the previous paragraph to get a new iterate $x_{[k|k]}^{(2)}$. When the iteration converges, we finally obtain the Huber's M-estimate at epoch k: $x_{[k|k]} = [z_{1|k}^{\mathrm{T}}, \ldots, z_{k|k}^{\mathrm{T}}, a_k^{\mathrm{T}}]^{\mathrm{T}}$. Then we start the computation for epoch $k + 1$.
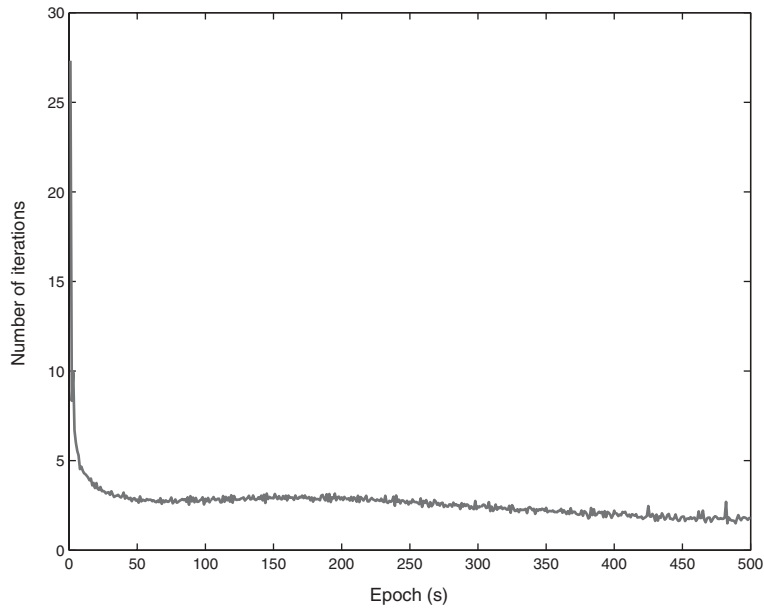
Fig. 3 Average number of iterations for the modified Newton method using simulated code and carrier-phase data
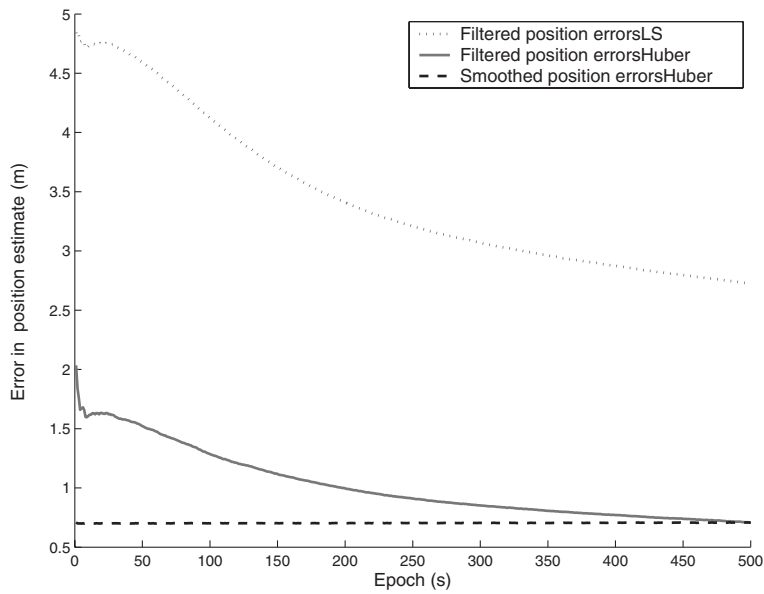

Fig. 4 Average errors in the position estimates using simulated code and carrier-phase data

Remark In Eq. (33), all measurement equations from epoch 1 to epoch k are used for the estimation at epoch k. The computation cost at each epoch increases when k increases, so does the computer storage requirement. This may cause problems, especially for real-time GPS applications, when k becomes too large. In practice, after certain epochs, we may use only a fixed number of latest measurements for positioning. Our method with slight modification can still be applied. At each new epoch, we add the new measurement equations to the model and discard the oldest ones. When we compute the search direction in the first iteration at the new epoch, we

just simply discard the part related to the oldest measurement equations in each matrix and vector in Eq. (48). This does not cause any difficulty in computations.

5.2 Simulation results

To demonstrate the performance of our method, we give some simulation results here. The simulation scenario is the same as that described in Sect. 4. The code measurements were

constructed as Eq. (28) and carrier-phase measurements were constructed as follows:

$$y_k^\phi = \bar{y}_k + e\beta_k + a + v_k^\phi + b^\phi, \qquad (53)$$

where $\bar{y}_k$ and β_k are the same as given in Eq. (28), the ith element of a is λN^i with N^i being the difference of the number of integer cycles between satellite i and the roving receiver and that between satellite i and the stationary receiver at epoch 1, $v_k^\phi \sim \mathcal{N}(0, \sigma_\phi^2 I)$ with $\sigma_\phi = 0.01$ m, and b^ϕ is the outlier vector.

We used the modified Newton method to compute Huber's M-estimates of the positions. In each epoch, the iteration process stops when the difference between the position estimates at two consecutive iteration steps is less than 0.001 m. The tuning constant γ was set to be $1.5\sigma_\phi$. For comparison, we also computed the LS estimates of the positions by a recursive LS method proposed in Chang and Paige (2003).

We used eight visible satellites and took the outlier vectors $b^\rho = [0, 10, 0, 0, 0, 0, 0, 0]^T$ and $b^\phi = [0, 0.1, 0, 0, 0, 0, 0, 0]^T$ (in meters) for each epoch. One exception was that b^ϕ was not added to the single-differenced carrier-phase measurements in the first epoch, since otherwise the Huber's M-estimation could not reduce the influence of the outliers in the carrier-phase measurements for the reason we mentioned in Sect. 5.1. In order to reduce random effect on the results, we performed 100 simulation runs for the same satellite geometries, and each run had 500 epochs.

Figure 3 displays the *average* number of iterations for the modified Newton method. For the initial epochs, the number of iterations is relatively large, but it drops dramatically later. For most epochs, it takes only two to three iterations, indicating that modifying the Newton search directions for efficiency and for handling possible singularity problems had no negative effect on the convergence for those epochs.

Figure 4 displays the *average* filtered position errors in Huber's M-estimate $x_{k|k}$ and in the LS estimate for $k = 1, \ldots, 500$. As for the code-based relative positioning, we observe that Huber's M-estimation can reduce the effect of outliers and give better position estimates than the LS estimation. Both errors tend to decrease with increasing epoch. Figure 4 also gives the *average* errors in the smoothed position estimates $x_{k|500}$ for $k = 1, \ldots, 499$. As expected, the smoothed position estimates are better than the corresponding filtered position estimates, since more information was used by the former. We see that the smoothed position estimates at different epochs have almost the same accuracy. This is because the same estimate of the ambiguity vector was involved in the estimation of these positions.

6 Summary

The IRLS method is often used in geodesy for robust M-estimation. For Huber's M-estimation, we proposed to use Newton's method instead, since it is usually more computationally efficient than the IRLS method. We showed by simulations that for code-based differential positioning, Newton's

method converges faster than the IRLS method. For code- and carrier-phase-based relative positioning, we showed how to use the structures of the model to design an efficient modified Newton method for recursively computing Huber's M-estimates. When we designed the numerical method, we also took numerical reliability and storage requirement into account. Simulation results indicate that the proposed numerical methods are effective.

We only considered the computation of position estimates in this paper. In order to know how accurate the position estimates are, we need to compute the variance–covariance matrices of the estimates. This will be studied in the future. In real applications, the observations are correlated to some extent. We have to study how to deal with this problem in our method. Comparisons of Huber's M-estimation method with other robust estimation methods and the fault detection and identification methods also need to be systematically investigated.

References

Axelrad P, Brown RG (1996) GPS navigation algorithms. In: Parkinson BW, Spilker JJ, Axelrad P, Enge P (eds) Global positioning system: theory and applications, vol 1. American Institute of Aeronautics and Astronautics Inc, pp 409–433

Antoch J, Ekblom H (1995) Recursive robust regression: computational aspects and comparison. Comput Stat Data Anal 19:115–234

Awange JL, Aduol FWO (1999) An evaluation of some robust estimation techniques in the estimation of geodetic parameters. Surv Rev 35:146–162

Björck Å (1996) Numerical methods for least squares problems. SIAM, Philadelphia

Brown RG (1996) Receiver autonomous integrity monitoring. In: Parkinson BW, Spilker JJ, Axelrad P, Enge P (ed) Global positioning system: theory and spplications, vol 2. American Institute of Aeronautics and Astronautics Inc, pp 143–165

Chang X-W (2005) Computation of the Huber M-estimates for a block-angular regression problem. Comput Stat Data Anal (in press)

Chang X-W, Paige CC (2003) An algorithm for combined code and carrier phase based GPS positioning. BIT Numer Math 43:915–927

Chang X-W, Paige CC, Yin L (2004) Code and carrier phase based short baseline GPS positioning: computational aspects. GPS Solut 7:230–240

Fuchs H (1982) Contribution on the adjustment by minimizing the sum of absolute residuals. Manuscr Geod 7:151–207

Gui Q, Zhang J (1998) Robust biased estimation and its applications in geodetic adjustments. J Geod 71:430–435

Huber PH (1964) Robust estimation of a location parameter. Annals Math Statist 35:73–101

Huber PH (1981) Robust statistics. Wiley, New York

Krarup T, Juhl J, Kubik K (1980) Götterdämmerung over least squares adjustment. In: Proceedings 14th congress of the international society of photogrammetry, vol B3, Hamburg, pp 369–378

Kelly RJ (1998) The Linear model, RNP, and the near-optimum fault detection and exclusion algorithm. In: Global positioning system, vol 5. The Institute of Navigation, Alexandria, pp 227–259

Koch KR (1999) Parameter estimation and hypothesis testing in linear model, 2nd edn. Springer, Berlin Heidelberg New York

Koch KR, Yang Y (1998) Robust Kalman filter for rank deficient observation models. J Geod 72:436–441

Madsen K, Nielsen HB (1990) Finite algorithms for robust linear regression. BIT 30:682–699

Misra P, Enge P (2001) Global positioning system, signals, measurements, and performance. Ganga-Jamuna Press, Lincoln

Nocedal J, Wright SJ (1999) Numerical optimization. Springer, Berlin Heidelberg New York

O'Leary DP (1990) Robust regression computation using iteratively reweighted least squares. SIAM J Matrix Anal Appl 11:466–488

Osborne MR (1985) Finite algorithms in optimization and data analysis. Wiley, Toronto

Rieder H (ed) (1996) Robust statistics, data analysis and computer intensive methods proceedings of the conference in honor of Peter Huber's 60th birthday. Lecture Notes in Statistics (109). Springer, Berlin Heidelberg New York

Teunissen PJG (1990) Quality control in integrated navigation systems. IEEE Trans Aerosp Electr Syst 5:35–41

Tiberius CCJM (1998) Recursive data processing for kinematic GPS surveying. PhD Thesis, Department of Mathematical Geodesy and Positioning, Delft University of Technology

Tiberius CCJM, Kenselaar F (2003) Variance component estimation and precise GPS positioning: case study. J Surv Eng 129:11–18

Wieser A, Brunner FK (2002) Short static GPS sessions: robust estimation results. GPS Solut 5:70–79

Wolke R (1992) Iteratively reweighted least squares: a comparison of several single step algorithms for linear models. BIT 32:506–524

Xu P (1989) On robust estimation with correlated observations. Bull geod 63:237–252

Yang Y (1991) Robust Bayesian estimation. Bull Geod 65:145–150

Yang Y (1994) Robust estimation for dependent observations. Manuscr Geod 19:10–17

Yang Y, He H, Xu G (2001) Adaptively robust filtering for kinematic geodetic positioning. J Geod 75:109–116

Zhong D (1997) Robustness estimation and optimal selection of polynomial parameters for the interpolation of GPS geoid heights. J Geod 71:552–561

Zhu J (1996) Robustness and the robust estimate. J Geod 70:586–590