

Published in:

Transportation Networks: Recent Methodological Advances, M.Bell ed.

IMPROVED ALGORITHMS FOR CALIBRATING GRAVITY MODELS

*Nanne J. van der Zijpp*¹

*Transportation and Traffic Engineering Section
Delft University*

Benjamin G. Heydecker

*Centre for Transport Studies
University College London*

-Abstract- Knowledge of Origin-Destination (OD) trip matrices is needed in many stages of transport planning. As direct observations of OD-flows are usually not available, a commonly used approach is to estimate OD-matrices from traffic counts. The under-specification that characterizes this estimation problem can be resolved by requiring that OD-matrices conform to a model of travel demand. In this paper we consider a class of methods based on the gravity model, with either an exponential or a discretized deterrence function. We focus on solution methods and establish a unifying analysis of them. A key result presented in the paper is an expression for the matrix of second derivatives of the likelihood objective function. Based on this result, a minimization method is developed with a greatly improved speed and convergence. Numerical experiments based on a set of empirical data obtained from a French tollroad have shown that speed has improved by a factor that typically is in the range 6 - 15 and can in some cases be as large as 400 relative to existing methods. The new method has good robustness in that it converges successfully under a wide variety of circumstances including those under which other methods fail. Moreover, this result provides an effective convergence criterion that can also be applied in combination with other solution methods, such as Gauss-Seidel and gradient search methods.

1. A large part of the work on which the paper is based was done while the first author stayed at the Center for Transport Studies, University College London

INTRODUCTION

The paper deals with the calibration of gravity models as a means to estimate Origin-Destination (OD) trip matrices from aggregated data sources such as traffic counts and trip-end constraints. Generally, many OD-matrices can be constructed that reproduce a given set of observations. Therefore the under-specification that characterizes the OD-estimation problem needs to be resolved. This can be done by requiring that OD-values conform to a model of travel demand. Several approaches of this kind have been discussed in the literature and are being used in practice. In the following we consider a class of methods based on the gravity model, with either an exponential or a piecewise constant deterrence function. We focus on numerical methods to calibrate these models from traffic counts and establish a unifying analysis of them.

In the past it has been shown that the gravity model can be underpinned in a number of ways, see e.g. Erlander and Stewart (1990). We confine ourselves here to specifying the gravity model and the related distributions of both model residuals and observation errors. A consequent likelihood expression is established which we seek to maximise. We concentrate on methods that can be used for this purpose. Existing methods are discussed, and it is shown that some of the difficulties that occur when applying them can be resolved by a change of variables to logarithms of the model parameters. A key result is the derivation of the matrix of second derivatives of the likelihood function. This result is then applied and leads to a new optimisation method. In a series of experiments this method is shown to be robust and computationally efficient.

MODELLING

Model Description

The description of the gravity model that is used here is given by:

$$\begin{aligned}
 T_{ij} &= T_{ij}^m + \epsilon_{ij} \\
 T_{ij}^m &= O_i D_j f(c_{ij}) \\
 \epsilon_{ij} &\sim N[0, T_{ij}^m] \\
 i &= 1, 2, \dots, I, \quad j = 1, 2, \dots, J
 \end{aligned} \tag{1}$$

where:

T_{ij}	OD-flow from zone i to zone j
I, J	number of origins, number of destinations
T_{ij}^m	model estimated OD-flow
ϵ_{ij}	modelling error
O_i, D_j	production ability, attraction ability
$f(\cdot)$	deterrence function
c_{ij}	generalized travel costs

We use a normally distributed error term with variance equal to the associated mean as moti-

vated by a Poisson process for the traffic counts. Various functional forms for the deterrence function $f(\cdot)$ can be used; see e.g. *Ortuzar and Willumsen (1994)* for an overview. Two frequently used classes of deterrence functions are considered:

Case 1: The exponential deterrence function

$$f(c_{ij}) = \exp[-\alpha c_{ij}] \quad (2)$$

Where α is the dispersion parameter.

Case 2: The piecewise constant deterrence function

$$f(c_{ij}) = \sum_{h=1}^H F_h \delta_h(c_{ij}) \quad (3)$$

where H is the number of cost ranges, F_h is the range specific cost, and $\delta_h(c_{ij})$ is given by:

$$\delta_h(c_{ij}) = \begin{cases} 1, & \text{if } c_{ij} \text{ is in cost range } h. \\ 0, & \text{otherwise.} \end{cases}$$

Traffic Counts

The model parameters are to be estimated from traffic counts or other aggregated flow observations. In our analysis, traffic counts are summarised in a vector y and are assumed to be linear combinations of OD-flows:

$$y = A T + \xi \quad (4)$$

with:

y	vector of traffic counts (with K elements)
T	vector of OD-flows.
$x(i,j)$	ordering of the matrix elements T_{ij} in the vector T .
a_{ijk}	proportion of flow from i to j that traverses link k
A	matrix of assignment proportions, $A_{k,x(i,j)} = a_{ijk}$, $k=1,2,\dots,K$
ξ	vector of counting errors

For the counting error ξ a zero mean multivariate normal (MVN) distribution with covariance matrix Ω is assumed. It should be noted that the above definition of traffic counts allows for a wide variety of data to be used, including trip-ends, screenline counts, and even survey data and prior matrices.

Maximum Likelihood Approach to Model Calibration

As a result of the assumptions given in (1) the conditional distribution of OD-vector T , given the

model parameters is given by:

$$T \sim \text{MVN}[T^m(O,D,F), \mathbf{T}^m] \quad (5)$$

with:

$T^m(O,D,F)$ vector of model predicted OD-flows
 O, D, F vectors of production abilities, attraction abilities and deterrence function parameter(s) respectively.
 $\mathbf{T}^m$ diagonal matrix with the elements of T^m on its main diagonal.

i.e. the distribution of T is given by a multivariate normal distribution with expected value T^m , and covariance matrix $\mathbf{T}^m$. Combining equations (5) and (4) results in an expression for the conditional distribution of the traffic counts given the model parameters:

$$y \sim \text{MVN}(A T^m, A \mathbf{T}^m A' + \Omega) \quad (6)$$

where T^m should satisfy the requirements that follow from (1). In theory, the parameters O, D and F can be found by maximising the joint likelihood. In that case these parameters are given by:

$$\begin{bmatrix} O \\ D \\ F \end{bmatrix} = \underset{O, D, F}{\text{argmax}} \text{MVN}(A T^m, A \mathbf{T}^m A' + \Omega) \Big|_y \quad (7)$$

Simplifying the Likelihood Objective Function

In (7) both the expectation and the covariance matrix of the density that is to be maximised depend on the model parameters. In the present paper we discuss algorithms that can be used to maximise the likelihood if the covariance matrix is approximated with a fixed covariance matrix Σ . The approximation Σ may be obtained by substituting a prior estimate of the OD-flows into equation (7) or by assuming that Σ is a diagonal matrix with the observed values y on its diagonal. In Van Der Zijpp and Heydecker (1996b) an iterative extension of these algorithms is described that can be used solve (7). The problem with the fixed covariance matrix corresponds to:

$$\begin{aligned} & \underset{O \geq 0, D \geq 0, F \geq 0}{\text{minimize}} && J(O,D,F) = (y - A T^m)' \Sigma^{-1} (y - A T^m) \\ & && T^m = T^m(O,D,F) \end{aligned} \quad (8)$$

There are several advantages in formulating the OD-estimation in this way:

- The traffic counts that are used may represent arbitrary linear combinations of OD-cells,
- The objective function offers the possibility to calibrate a gravity model, even if no explicit data are available on trip ends and/or trip length distribution,
- The possibility exists to specify the variance of counting errors, or to weigh the observations according to the confidence in them,

- Inconsistencies in traffic counts as a result of error need not be removed prior to calibration,
- Spatial dependencies that occur due to the fact that the model prediction error, ε_{ij} , is observed at several locations can be accommodated (see Van Der Zijpp and Heydecker, 1996b).

EXISTING ALGORITHMS FOR CALIBRATING THE GRAVITY MODEL

The method implied by equation (8) has a number of desirable properties such as its flexibility with respect to input data and the possibility to weigh various data items according to their assumed accuracies. However, a problem is that in order to solve (8) an algorithm is needed for a highly non-linear minimization problem subject to non-negativity constraints. The absence of an algorithm that performs this task under a sufficiently wide variety of circumstances might very well explain the relative unpopularity of this method compared to, for example, entropy maximising methods. This section discusses a number of numerical methods that can be used to solve (8). For a more elaborate overview, see Van Der Zijpp and Heydecker (1996b).

Balancing Methods

The *bi-proportional method* starts with an initial or prior matrix. This matrix is subsequently scaled row by row and then column by column to yield the observed or given trip end totals. The iterations stop if no more substantial changes occur in the estimated matrix, i.e. when the observations are reproduced closely. An equivalence with the gravity model with exponential deterrence function exists if in the initial tableau the cells are replaced with values derived from an exponential deterrence function. Note that the bi-proportional method cannot be used to find appropriate values for α in (2). Instead Hyman's method (Hyman, 1969) can be used to estimate α . A prerequisite to apply Hyman's method is that the average trip length is known.

Over and above the assumptions of the bi-proportional method, in the *tri-proportional method* the prior matrix is scaled to force the estimated matrix to match an observed trip length distribution (OTLD). An equivalence with the gravity model with piecewise constant deterrence function is reached if the prior matrix is replaced with a matrix of ones.

In a balancing method, the iterations continue until the estimated matrix reproduces all observations. A requirement for convergence of the bi- and tri-proportional balancing methods is that such a solution exists, i.e. that the traffic counts that are used are mutually consistent. The analogy between balancing methods and other gravity model calibrating methods disappears if an initial matrix is scaled to meet measurements that do not represent trip ends, as is done in entropy maximizing approaches such as ME2 (Ortúzar and Willumsen, 1994; Willumsen, 1991; Van Zuylen and Willumsen, 1980). Similarly, if an initial tableau is derived from a prior matrix the solution resulting from applying a balancing method no longer satisfies the gravity model.

Balancing methods can hence only be applied to find a candidate solution to (8) in cases where:

- observations and constraints represent trip ends or observed trip length distributions, and
- observations are mutually consistent.

In this case, the solution that will be found will reproduce the observations exactly and will hence correspond to a zero value of the objective function (8), which is its global minimum. Balancing methods are popular due to their ease of implementation. However, convergence can be slow (see Maher, 1983) if not uncertain. Proofs of convergence apply only to special cases, e.g. only trip-ends are given and the deterrence function is kept constant, see Evans (1971) and Kirby (1974). In the more general case where observations represent traffic counts and may not be mutually consistent, other techniques should be considered, two of which are discussed below.

Gradient Search Methods

If we consider a model with a piecewise constant deterrence function, see (1) and (3) it is possible to find an analytical expression for the derivatives of the objective function (8) with respect to the model parameters O_i , D_j , F_h and α . The resulting gradient vector can be used as a basis for an iterative scheme to minimise (8). The simplest way this can be done is to apply a steepest descent method. However, steepest descent method tend to converge very slow. A gradient search methods with a higher speed of convergence is the conjugate gradient method, see Bazaraa et al (1993). An easy way to find conjugate gradient search directions was proposed by Fletcher and Reeves (1964). There are however a number of difficulties that occur when a conjugate gradient method based on Fletcher and Reeves is applied to minimise (8):

- The solution to the line searches cannot be expressed in an analytical way. Hence, the line searches must be done numerically,
- The parameter space is constrained to non-negative values. This makes the line search more complex, and it also invalidates the Fletcher-Reeves method to find conjugate gradients,
- It is difficult to establish an effective convergence criterion,
- Convergence can be extremely slow.

These factors make it difficult to construct a gradient search method that operates in a reliable way under all circumstances.

Gauss-Seidel Methods

Another way to tackle the minimization problem (8) is to divide the model parameters in groups and to solve the parameters in each group separately, while keeping the parameters in other groups temporarily constant. This iterative process is continued until a fixed point is found. For problem (8) it seems advantageous first to solve the vector of production abilities (O), then to solve the vector of attraction abilities (D), and then to solve the vector of deterrence function values (F). This procedure can only be used to calibrate a gravity model with a piecewise constant deterrence function (case 2). Each step in this sequence corresponds to minimizing a quadratic function. If the nonnegativity constraints can be ignored, this can be solved at the expense of one matrix inversion, so computationally costly numerical line searches can be avoided. The details of the method are described in Van Der Zijpp and de Romph (1996).

Although the method works well in a large number of cases and is easy to implement with the

help of modern software packages, there are a number of technical difficulties:

- No theoretical proof of convergence exists,
- The minimization problems can be expensive to solve if the nonnegativity constraints come into play,
- The method does not calibrate gravity models with exponential deterrence functions,
- It is difficult to establish an effective convergence criterion.

A UNIFYING ANALYSIS OF THE ML-OBJECTIVE FUNCTION

We now show how, by using a logarithmic transformation, the gravity model formulations presented here can be unified and solved conveniently.

It has been shown earlier (see e.g. Sen and Smith, 1995) that both the gravity model with exponential deterrence function (equations (1) and (2)), and the gravity model with piecewise constant deterrence function (equations (1) and (3)) can be summarized in the following form:

$$T^m(O,D,F) = t(q) = \exp(Cq) \quad (9)$$

where

$\exp(.)$	is the exponentiation operator, applied element by element
q	is the vector of logarithms of the model parameters, i.e.
$q = \log \begin{pmatrix} O \\ D \\ F \end{pmatrix}$	
C	is a matrix with three nonzero elements per row. The first and second nonzero elements are given by respectively:

$$\begin{aligned} C_{x(i,j),i} &= 1 \\ C_{x(i,j),I+j} &= 1 \end{aligned} \quad (10)$$

The way the third nonzero element is defined depends on whether the deterrence function is specified to be exponential or piecewise constant:

Case 1: The exponential deterrence function

$$C_{x(i,j),I+J+1} = -c_{ij} \quad (11)$$

Case 2: The piecewise constant deterrence function

$$C_{x(i,j),I+J+h} = \delta_h(c_{ij}) \quad , \quad h=1,2,\dots,H \quad (12)$$

In summary, the problem of estimating OD-matrices from traffic counts using a gravity model combined with either an exponential or a piecewise constant deterrence function is equivalent to the minimization of the following objective function:

$$\begin{aligned} J(q) &= (y - A t(q))' \Sigma^{-1} (y - A t(q)), \\ t(q) &= \exp(Cq) \end{aligned} \quad (13)$$

Compared to the formulation given in (8) this change of variables has the following practical advantages:

- A single objective function can be used to represent methods based on exponential and piecewise constant deterrence functions,
- The minimization is not subject to nonnegativity constraints,
- Ease of notation and analysis. This advantage will become apparent in the next section, where expressions for the gradient and the Hessian (the matrix of second derivatives) of the objective function with respect to the vector q will be derived.

DERIVING THE GRADIENT AND HESSIAN OF THE OBJECTIVE FUNCTION

Definitions, Lemmas and Proofs

For a scalar valued function f of a vector valued variable x we will denote the *gradient* and the *Hessian* with respect to x with the symbols $\nabla_x[f(x)]$ and $\nabla_x^2[f(x)]$ respectively. Note that the Hessian of f , i.e. the matrix of second derivatives of f , is the *Jacobian* of the gradient of f , i.e.:

$$\nabla_x^2[f(x)] = \nabla_x[\nabla_x[f(x)]] \quad (14)$$

where the outer ∇_x is applied elementwise. Define the $\text{diag}(\cdot)$ operator as the operator that transforms a vector into a diagonal matrix by putting its elements on the main diagonal. Furthermore, define the *array multiplication* with:

$$C = A \otimes B \Leftrightarrow C_{ij} = A_{ij} B_{ij}, \forall i, j \quad (15)$$

In connection with these definitions, the following lemma's are proven:

Lemma 1: for any two vectors a and b and matrix C of appropriate size, the following holds:

$$\text{diag}(a) C \text{diag}(b) = (a b') \otimes C \quad (16)$$

Proof (Lemma 1)

$$D = \text{diag}(a) C \text{diag}(b) \Leftrightarrow D_{ij} = a_i b_j C_{ij}, \forall i, j \Leftrightarrow D = (a b') \otimes C \quad (17)$$

■

Lemma 2: let f be a vector valued function of x of appropriate size, then:

$$\nabla_x[\text{diag}(x) f(x)] = \text{diag}(f(x)) + \nabla_x[f(x)] \text{diag}(x) \quad (18)$$

Proof (Lemma 2):

According to lemma 1:

$$\text{diag}(x) f(x) = \text{diag}(x) f(x) \text{diag}(1) = x \otimes f(x) \quad (19)$$

Hence:

$$\frac{\partial (\text{diag}(x) f(x))_j}{\partial x_i} = \frac{\partial x_j}{\partial x_i} f_j(x) + x_j \frac{\partial f_j(x)}{\partial x_i} \quad (20)$$

Or, in matrix notation, with e being a vector of ones, and lemma 1 applied once more:

$$\begin{aligned} \nabla_x [\text{diag}(x) f(x)] &= \nabla_x [x] \otimes (f(x)e') + (e x') \otimes \nabla_x [f(x)] \\ &= I \otimes (f(x)e') + \text{diag}(e) \nabla_x [f(x)] \text{diag}(x) \\ &= \text{diag}(f(x)) + \nabla_x [f(x)] \text{diag}(x) \end{aligned} \quad (21)$$

■

Computing the Gradient of $J(q)$

The gradient of the objective function $J(q)$ defined in (13) satisfies:

$$\nabla_q J(q) = (\nabla_q t(q)) \nabla_t J(t) \quad (22)$$

Referring to equation (13), we substitute $\nabla_q t(q) = C' \text{diag}(t)$ and $\nabla_t J(t) = 2A' \Sigma^{-1}(At-y)$ in (22). This yields:

$$\nabla_q J(q) = 2C' \text{diag}(t) A' \Sigma^{-1}(At-y) \quad (23)$$

Computing the Hessian of $J(q)$

Using equation (23) it can be seen that the Hessian of $J(q)$ satisfies:

$$\begin{aligned} \nabla_q^2 [J(q)] &= \nabla_q [\nabla_q J(q)] \\ &= \nabla_q [2C' \text{diag}(t) A' \Sigma^{-1}(At-y)] \\ &= \nabla_q [t(q)] \nabla_t [2C' \text{diag}(t) A' \Sigma^{-1}(At-y)] \\ &= C' \text{diag}(t) \nabla_t [2C' \text{diag}(t) A' \Sigma^{-1}(At-y)] \end{aligned} \quad (24)$$

Let $g(t) = \text{diag}(t) A' \Sigma^{-1}(At-y)$, then:

$$\begin{aligned} \nabla_q^2 [J] &= C' \text{diag}(t) \nabla_t [2C' g(t)] \\ &= C' \text{diag}(t) \nabla_t [g(t)] \nabla_g [C' g] \\ &= 2C' \text{diag}(t) \nabla_t [\text{diag}(t) A' \Sigma^{-1}(At-y)] C \end{aligned} \quad (25)$$

Now we evaluate $\nabla_t [\text{diag}(t) A' \Sigma^{-1}(At-y)]$. Applying lemma 2 gives:

$$\begin{aligned} \nabla_t [\text{diag}(t) A' \Sigma^{-1}(At-y)] &= \text{diag}(A' \Sigma^{-1}(At-y)) + \nabla_t [A' \Sigma^{-1}(At-y)] \text{diag}(t) \\ &= \text{diag}(A' \Sigma^{-1}(At-y)) + A' \Sigma^{-1} A \text{diag}(t) \end{aligned} \quad (26)$$

Hence:

$$\nabla_q^2 [J(q)] = 2C' [\text{diag}(t) \text{diag}(A' \Sigma^{-1}(At-y)) + \text{diag}(t) A' \Sigma^{-1} A \text{diag}(t)] C \quad (27)$$

Equations (23) and (27) give insight into the performance of existing solution methods, and provide a foundation for new ones.

Implications for Existing Methods

The gradient search methods and Gauss-Seidel methods discussed in this paper can be used effectively to find stationary points of an objective function. Whether such a point represents a local minimum or a saddle point can easily be determined by evaluating the Hessian at that point: a local minimum only exists if the Hessian is positive definite (PD). Therefore result (27) can be used as a useful addition to existing methods: it supplies a powerful convergence criterion. However the analysis needed to determine if a particular local minimum to objective function (13) is also a global minimum is more intricate; this problem is still open to investigation.

A NEW ALGORITHM TO CALIBRATE GRAVITY MODELS

It is possible to define a new numerical method to minimize (13) using expression (27) for its Hessian. This method is based on a quadratic approximation of the objective function of the form:

$$J^*(q+\epsilon)=J(q) + \epsilon' \cdot \nabla_q [J(q)] + \frac{1}{2} \epsilon' \nabla_q^2 [J(q)] \epsilon \quad (28)$$

This expression can be minimised over the vector ϵ at the cost of one matrix inversion, provided that the Hessian of J evaluated at point q is PD. An iterative method based on this strategy is known as the Newton method. The advantage of this method is that each successive iterate can be found analytically, so no numerical line search procedure is needed. Two factors should be taken into account when applying this method to minimize objective function (13):

- The Hessian $\nabla_q^2 [J(q)]$ need not necessarily be strictly PD for all q . Therefore the quadratic approximation of the objective function, J^* , may not be bounded below. In cases where it is not, the next iterate would not be defined. Therefore a modified Newton will be used to ensure that each successive iterate exists and will reduce the objective function. In the present paper, we adopt the Levenberg-Marquardt type of method, see Bazaraa et al (1993), p312. This method will be explained below.
- The vector q contains the logarithms of the model parameters. This ensures that the nonnegativity requirements are satisfied. However there does not exist a bounded value of q that corresponds to a solution to (8) that contains one or more zero parameter values. Although the value of zero can be approximated arbitrarily closely, it may not be possible to represent such small numbers internally in a computer, resulting in problems of convergence caused by rounding errors or underflow. Therefore the following penalty function is added to the objective function:

$$f(q) = c_1 \sum_i \max(0, -c_2 - q_i)^3 \quad (29)$$

This function is continuous in its first and second derivatives, and prevents the problems mentioned above. The values that were chosen for c_1 and c_2 were 100 and 20 respectively.

The idea behind the Levenberg-Marquardt method is that even if the Hessian in a particular point is not PD, a next iterate with a lower objective function value can be found by performing a small step in the direction of the negative gradient. This is effected by adding a multiple μ of the identity matrix $\mathbf{I}$ to the Hessian in (28), before minimizing this expression. The values of μ are modified between iterations using a strategy that is illustrated in figure 1. The next value of μ depends strongly on the ratio R of the true improvement and the predicted improvement. If this ratio is favourable then the value of μ is decreased; otherwise it is increased. The thresholds that were used are given in Bazaraa et al (1993).

NUMERICAL EXPERIMENTS

Numerical experiments have been performed on a set of empirical trip data obtained from a French toll road. These data consist of an observed trip table for eight different days between 29 origins and 26 destinations, resulting in 570 OD-flows that contribute to link-flows observed in the study area, and 107 traffic volumes have been synthesized from them. The variance of each of these traffic counts was taken to be equal to its observed value. There is no route choice in this network because it is linear. Two types of experiments were undertaken.

The first series of experiments involved estimating an OD-matrix by calibrating a gravity model on the traffic counts. This was done for a model with a piecewise constant deterrence function, see eq. (3). The number of distance classes that were distinguished was increased progressively from 1 to 16. Each additional class was formed by sub-dividing an existing one, so that a hierarchical sequence of models was constructed.

From the methods discussed in this paper, the conjugate gradient, Gauss-Seidel and modified Newton method were implemented to minimise the objective function. The conjugate gradient search method turned out to be extremely slow, which prevented the completion of a full set of experiments for this method. In table 1 the results were summarized for the Gauss-Seidel method and the modified Newton method. Each row in this table contains the results of calibrating the model using a deterrence function with the corresponding number (N) of classes, averaged over the eight different days for which observed data were available. The model prediction error could be computed by comparing the model estimated OD-flow with the value contained in the toll-ticket database, and is expressed using the Root Mean Squared Error (RMSE). The log-likelihood value is denoted with L , and decreases with the addition of each extra class. The other columns in the table contain the average CPU-time (on a Pentium 90) and the number of cases for which the algorithm did not converge before a preset maximum number of 2000 iterations was reached (indicated with the symbol E).

In all cases, a solution with an equal or higher likelihood value was computed with the modified Newton method. Moreover, with this method a solution that satisfied the convergence criterion was reached without exception, whilst the Gauss-Seidel exceeded the maximum number of iterations in 12 of the 128 cases. The main advantage of the modified Newton method is its speed: none of the cases that were considered took more than 290 seconds, whilst the Gauss-Seidel took 2975 seconds in one case; the average values were 106 and 626 seconds respectively. The experiments also revealed that for this particular dataset the best fit according to the RMSE

measure is obtained if a deterrence function with six classes is used. Unfortunately, in practice it is not possible to determine this number as one needs a completely observed trip matrix to compute this measure. This issue is discussed in Van Der Zijpp and Heydecker (1996a).

In a second series of experiments each of the 570 observed OD-flows was used as a measurement. This is similar to the case in which a prior matrix is available. When calibrating the gravity model on these data the solution with the highest likelihood value and that with the lowest RMSE coincide: the resulting RMSE is known as the *model specification error*, see Van Der Zijpp and Heydecker (1996a). The results of the calibration are shown in table 2. In general, the computation time needed for the Gauss-Seidel method was greater than the corresponding one in the first series, whilst the average computation time needed for the modified Newton method was substantially less. Again the Newton method converged in all cases, whilst the Gauss-Seidel did not meet the convergence criterion in 19 of the 128 cases. The maxima of the computation times over these 128 computations are 4671 seconds for the Gauss-Seidel method and 16.42 seconds for the Newton method; the average values were 1489 and 9.82 seconds respectively.

CONCLUSIONS

It has been shown that the problem of calibrating a gravity model is equivalent to the minimization of a highly non-linear objective function. This formulation has a number of advantages over more traditional ones, including the possibility to use arbitrary traffic counts and constraints, not requiring the presence of an observed trip length distribution (OTLD) and taking into account counting errors and spatial dependencies. However, the minimization of this objective function is intricate. Specific difficulties are the presence of the nonnegativity constraints, slow convergence, the absence of an adequate convergence criterion, and the complexity of implementation. The problem of the nonnegativity constraints can be avoided by a change of variables to the logarithms of the model parameters. This formulation unites methods with exponential and piecewise constant deterrence functions. A key result presented in the paper is an expression for the matrix of second derivatives of the objective function. Based on this matrix, a minimization method is presented with a greatly improved speed of convergence. Moreover, this result provides an effective convergence criterion that can also be applied in combination with other solution methods, such as Gauss-Seidel and gradient search methods.

-acknowledgements- This work was supported by an EC grant in the Human Capital and Mobility programme, contract ERBCHGCT930412

REFERENCES

- Bazaraa, M.S., Sherali, H.D. and Shetty, C.M. (1993) *Nonlinear Programming, Theory and Algorithms*, second edition, Wiley
- Erlander, E. and Stewart, N.F. (1990) *The Gravity Model in Transportation Analysis, Theory and Extensions*, VSP, Utrecht, the Netherlands
- Evans, A.W. (1971) The Calibration of Trip Distribution Models with Exponential or Similar

- Cost Functions, *Transportation Research*, Vol. 5, pp. 15-38
- Fletcher, R. and Reeves, C.M. (1964) Function Minimization by Conjugate Gradients, *The Comp. J.*, 7, pp. 149-154
- Hyman (1969) The Calibration of Trip Distribution Models, *Environment and Planning*, 1(3), pp.105-112
- Kirby, H.R. (1974) Theoretical Requirements for Calibrating Gravity Models, *Transportation Research*, Vol. 8. pp. 97-104
- Maher, M.J. (1983) Speeding up the Furness Method, *Traffic Engineering and Control*, pp.485-486
- Ortúzar, J. D. and Willumsen, L.G. (1994) *Modelling Transport*, Second Edition, JohnWiley and Sons, Chichester, England, ISBN 0 471 94193 X
- Willumsen, L.G. in: Papageorgiou, M. (1991) *Concise Encyclopedia of Traffic & Transportation Systems*, Pergamon Press, pp.317
- Sen, A, and Smith, T.O. (1995) *Gravity Models of Spatial Interaction Behavior*, Springer, ISBN 3-540-60026-4
- Van Der Zijpp, N.J. and Heydecker, B. (1996a) How many Parameters should a Traffic Model have?, *UTSG, ann.conference*, University of Huddersfield, unpublished
- Van Der Zijpp, N.J. and Heydecker, B. (1996b) Some Investigations into the Use of the Gravity Model, *Proc. 24rd European Transport Forum, PTRC*, Seminar D&E - Part 1, Brunel University, 2-6 september
- Van Der Zijpp, N.J. and de Romph E. (1996) A Dynamic Traffic Forecasting Application on the Amsterdam Beltway, *Int. Journal of Forecasting*, 1996, forthcoming
- Van Zuylen, H.J. and Willumsen, L.G. (1980) The most Likely Trip Matrix Estimated from Traffic Counts *Transportation Research-B*, Vol. 14B, pp. 281-293

**Table 1: Estimation results based on traffic counts
for the number of classes ranging from 1 to 16**

Gauss-Seidel					Newton			
N	RMSE	L	av.CPU	E	RMSE	L	av.CPU	E
1	537.33	-17443.93	321		536.91	-17443.90	77	
2	485.57	-11308.71	245		486.09	-11308.68	223	
3	470.82	-8184.56	255		470.81	-8184.55	151	
4	478.93	-7514.64	984		480.46	-7182.98	86	
5	387.47	-5176.33	506		387.45	-5176.31	166	
6	380.23	-4940.86	202		380.18	-4940.86	61	
7	381.40	-4934.20	113		381.46	-4934.19	37	
8	639.94	-4539.26	1181	3	643.58	-4538.52	94	
9	805.85	-4229.48	1648	5	806.40	-4227.91	64	
10	783.25	-3533.62	1099	2	784.37	-3533.28	150	
11	840.49	-3467.37	634		839.90	-3467.30	61	
12	840.47	-3467.37	56		839.90	-3467.30	33	
13	843.22	-3459.97	436		843.65	-3459.91	94	
14	872.92	-3427.43	439		874.67	-3427.38	51	
15	700.29	-2999.34	724		699.89	-2998.97	137	
16	777.60	-2719.70	1176	2	777.74	-2719.48	206	
tot.av.CPU			626					106
max. CPU			2975					289

**Table 2: RMSE fit of the gravity model
for the number of classes ranging from 1 to 16**

Gauss-Seidel				Newton		
N	RMSE	av.CPU	E	RMSE	av.CPU	E
1	329.36	307		329.36	14.82	
2	273.12	408		273.12	10.29	
3	235.05	946		235.05	11.08	
4	224.08	769		224.08	9.75	
5	216.94	807		216.94	10.38	
6	215.03	1191		215.03	8.23	
7	214.74	1392	1	214.74	7.83	
8	213.83	837		213.83	7.74	
9	212.45	1055		212.45	10.80	
10	212.37	1093	1	211.28	8.86	
11	207.30	1664	2	206.71	9.25	
12	205.97	1697	1	205.87	8.63	
13	203.40	1891	2	203.20	9.20	
14	202.53	2422	2	202.49	8.33	
15	196.48	4165	7	196.35	13.13	
16	196.23	3180	3	196.22	8.82	
tot.av.CPU		1489				9.82
max. CPU		4671				16.42

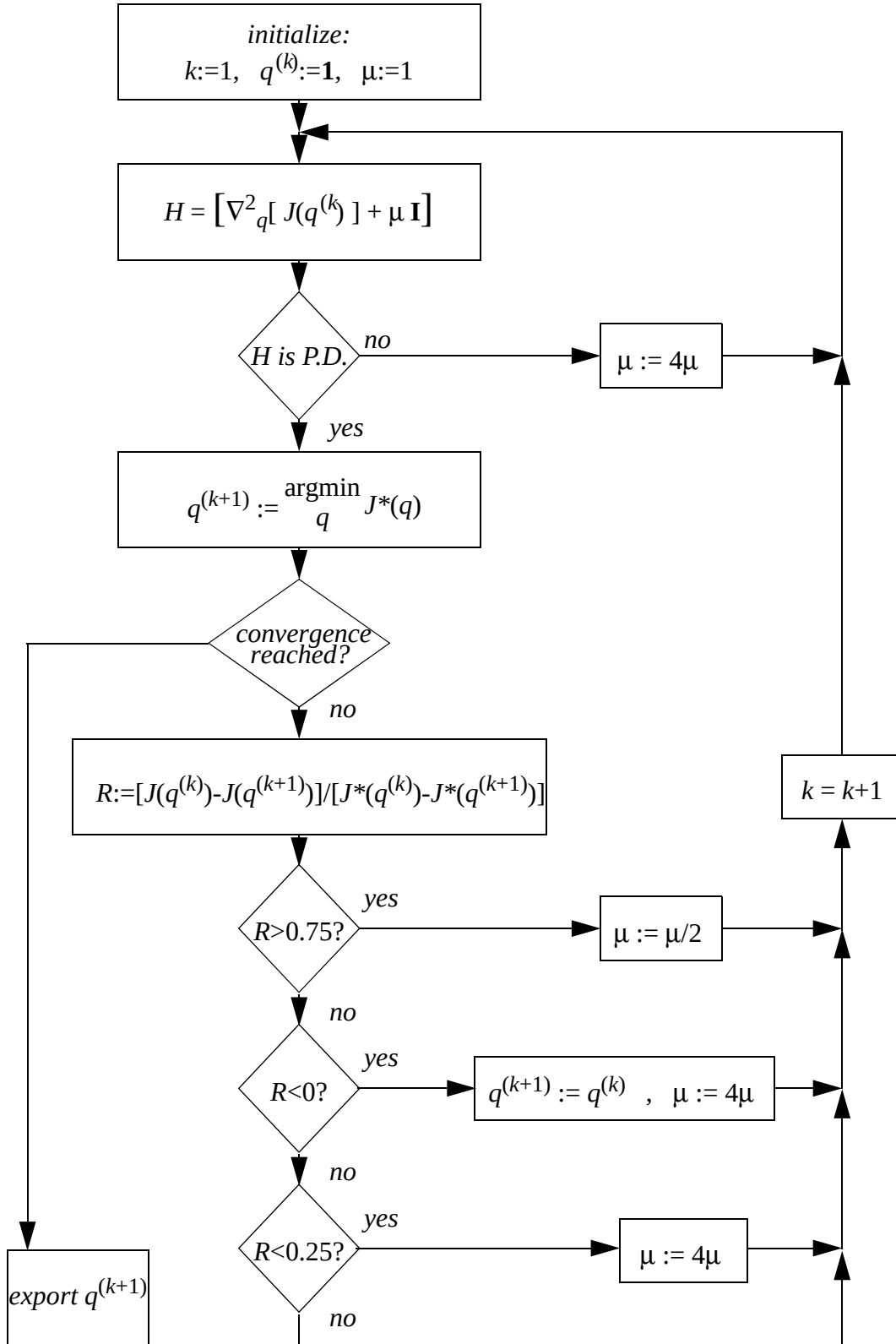


Figure 1: Overview of Levenberg-Marquardt method (modified Newton method)