The Process to Get Convergence of Balancing Factor in Gravity Model

Rahayu SULISTYORINI
Lecturer
Department of Civil Engineering
Lampung University
Jl. Sumantri Brojonegoro No.1
Phone&Fax: 062 – 0721 704947
E-mail: Sulistyorini_smd@yahoo.co.uk

Nindyo CAHYO KRESNANTO
Lecturer
Department of Civil Engineering
Janabadra University
Jl.Tentara Rakyat Mataram 57
Yogyakarta, Indonesia
Phone&Fax : 062 - 0274 543676
E-mail: nindyo_ck@yahoo.co.id

Abstract: The development of techniques for calibrating the trip distribution models to obtain O-D matrices is well advanced. Therefore, the study of transportation demand should be implemented with distribution of trips in origins and destinations. The first step to study trip distribution is to divide physical space under investigation into mutually exclusive zones or locations.

Calibration of the gravity model involves adjusting the friction factor with Newton-Raphson Method. An important consideration in developing the gravity model is "balancing" productions and attractions. Balancing means that the total productions and attractions for a study area are equal. The objective of this paper is to present the pattern of balancing factor in Gravity to achieve convergence with any beta values as input, using Matlab program. It is shown, by way of exponential as friction factor and using Double Constraint Gravity as balancing factor, we could see that the bigger beta value as input, the longer iteration to achieve convergence.

Key Words: *Gravity Model, Balancing Factor, Convergence*

1. INTRODUCTION

The transportation activity is essentially to link people and goods that are spatially separated. Calibration of the gravity model involves adjusting the friction factor with Newton-Raphson Method. An important consideration in developing the gravity model is "balancing" productions and attractions. Balancing means that the total productions and attractions for a study area are equal.

Sometimes, we put any beta value as first input to count balancing factor in modeling. The objective of this paper is to present the pattern of balancing factor in Gravity to achieve convergence with any beta values as input, using Matlab program. The friction factor is exponential function, and the type of balancing factor is DCGR. Iterative solution algorithms, that are modifications of the Newton-Raphson Technique, are proposed to solve each of the model formulations.

2. O-D MATRICES

The origin-destination matrix is a key element of the classical four-stages model, widely used in transportation modeling. This approach assumes that the survey area is divided into several zones, those are considered as indivisible entities during the modeling process.

OD matrix estimation, corresponding to the distribution phase of the four-stages process, is the most popular representation of transportation demand. It consists in defining a two-entries table, called the demand matrix or origin-destination (OD) matrix, whose rows and columns represent the zones of the study area. A cell of the matrix refers therefore to a particular origin-destination pair, and contains the total number of people accomplishing this journey.

This number has to be estimated from the data. Much research has concentrated on estimating an OD matrix and using it efficiently. Therefore, the study of transportation demand should be implemented with distribution of trips in origins and destinations. The first step to study trip distribution is to divide physical space under investigation into mutually exclusive zones or locations. Then, the trip distribution can be conveniently specified in a matrix form. If $t_{id}(k)$ indicates the number of trips during a given time period k between origin location i and destination location d, an n-by-n matrix $T(k) = [t_{id}(k)]$ can be constructed where n is the number of locations. This matrix is often called trip table, trip distribution matrix or origin-destination (O-D) matrix. In general, a typical O-D matrix is not symmetric and shown in **Figure 1.**

$$
\begin{array}{cc}
 & \textbf{\textit{Destination}} \\
\textbf{\textit{Origin}} & \begin{bmatrix} t_{11}(k) & t_{12}(k) & \cdots & t_{1n}(k) \\ t_{21}(k) & t_{22}(k) & \cdots & t_{2n}(k) \\ \vdots & \vdots & \ddots & \vdots \\ t_{n1}(k) & t_{n2}(k) & \cdots & t_{nn}(k) \end{bmatrix}
\end{array}
$$

Figure 1 O-D Matrices

From the O-D matrix, trip production and attraction of each location can be easily obtained. The trip production of i, $Ai(k)$ is the sum of row i representing the total number of trips originating from location i. The sum of column d, Bd (k) is the total number of trips destined to location d and is called the trip attraction of d. The grand total $T_{id}(k)$ of the O-D matrix, called total trip demand, represents the total number of trips across all locations. The following summarizes the characteristics of the O-D matrix.

3. GRAVITY MODEL

Gravity models are the most common form of trip distribution models currently in use. They are based on the assumption that the trip interchange between zones is directly proportional to the relative attractiveness of each zone, while inversely proportional to some function of the spatial separation between the zones. Gravity models have a number of theoretical advantages. However, they also have shortcomings. The most significant may be that they lack the ability to explain many behavioral aspects of destination choices, as travel costs and attraction are the only factors that solely determine the trip interchange patterns.

The gravity model is much like Newton's theory of gravity. The gravity model assumes that the trips produced at an origin and attracted to a destination are directly proportional to the total trip productions at the origin and the total attractions at the destination. The calibrating term or "friction factor" (f_{id}) represents the reluctance or impedance of persons to make trips of various duration or distances. The general friction factor indicates that as travel times increase, travelers are increasingly less likely to make trips of such lengths. Calibration of the gravity model involves adjusting the friction factor.

The socioeconomic adjustment factor is an adjustment factor for individual trip interchanges. An important consideration in developing the gravity model is "balancing" productions and attractions. Balancing means that the total productions and attractions for a study area are equal.

The analogous transport gravity model is:

$$T_{id} = k\frac{O_i O_d}{d_{id}^2} \quad : \quad k \text{ is a constant} \tag{1}$$

It shown that Trip produce at $\boldsymbol{i}$ an attracted and attracted at $\boldsymbol{d}$ equal as O_i and D_d , otherwise as square inverse to distance from $\boldsymbol{i}$ to $\boldsymbol{d}$. In mathematics form, this gravity model can be expresses as :

$$T_{id} = O_i D_d f_{id} \tag{2}$$

Equation (2) show if one of value O_i and one of value D_d become double, so trip between zone $\boldsymbol{i}$ to $\boldsymbol{d}$ increase four times. This is fatality error because the real trip is supposed to be increase two times. To solve this problems, we need a deterrence factor to limited equation T_{id}.

$$\sum_d T_{id} = O_i \text{ and } \sum_i T_{id} = D_d \tag{3}$$

Suppose now there are **M** modes travelling between zones, the modified gravity model (**Doubly-Constrained Gravity Model**) can then be expressed as:

$$T_{id} = \sum_m \left(O_i^m . D_d^m . A_i^m . B_d^m . f_{id}^m \right) \tag{4}$$

where: A_i^m and B_d^m = the balancing factors expressed as:

$$A_i^m = \left[\sum_d \left(B_d^m . D_d^m . f_{id}^m \right) \right]^{-1} \text{ and } B_d^m = \left[\sum_i \left(A_i^m . O_i^m . f_{id}^m \right) \right]^{-1} \tag{5}$$

Here, f_{id} is measure of accessibility from zone $\boldsymbol{i}$ to zone $\boldsymbol{d}$. Hyman (1969) stated there are three kinds of friction factor functions can be selected: power, exponential, and combined. The separation distance is referred to as the deterrence or impedance function and is often measured by travel time.

- Power Function : $f(C_{id}) = C_{id}^{-\alpha}$ (6)
- Exponential Function : $f(C_{id}) = e^{-\beta C_{id}}$ (7)
- Tanner/Combined Function : $f(C_{id}) = C_{id}^{\alpha} . e^{-\beta C_{id}}$ (8)

There are four kinds of *Gravity* model (Tamin,1997):

- Un Constraint Gravity (UCGR)
  This model is have no constraint, that's mean the total trip must be equal as Total trip that produced from production and attraction stage. Otherwise production and attraction from model is equal, but it shouldn't equal as production and attraction that we expected.

$$\boldsymbol{T_{id} = O_i A_i D_d B_d f_{id}} \ ,$$

As Equation 4, where $A_i = 1$ for all $\boldsymbol{i}$ and $B_d = 1$ for all $\boldsymbol{d}$**.**

- G*ravity* single constraint (Production Constraint Gravity/PCGR)
  Actual total trip from production is should be same as total trip as result from modeling. Production as result from modeling should be same as production we expected. Attraction from modeling could be different from attraction that expected.

  As Equation 4, where $\boldsymbol{B_d} = 1$ for all $\boldsymbol{d}$ and $\boldsymbol{A_i} = \dfrac{1}{\sum_d \left(\boldsymbol{B_d D_d f_{id}}\right)}$ for all $\boldsymbol{i}$.

- G*ravity* single constraint (Attraction Constraint Gravity/ACGR)
  Actual total trip from production is should be same as total trip as result from modeling. Attraction as result from modeling should be same as attraction we expected. Production from modeling could be different from production that expected.

  As Equation 4, where $\boldsymbol{A_i} = 1$ for all $\boldsymbol{i}$ and $\boldsymbol{B_d} = \dfrac{1}{\sum_i \left(\boldsymbol{A_i O_i f_{id}}\right)}$ for all $\boldsymbol{d}$

- Double Constraint Gravity (DCGR)
  Production and Attraction in this model should be same as expected, see equation 4..

$$\boldsymbol{A_i} = \frac{1}{\sum_d \left(\boldsymbol{B_d D_d f_{id}}\right)} \text{ for all } \boldsymbol{i} \text{ and}$$

$$\boldsymbol{B_d} = \frac{1}{\sum_i \left(\boldsymbol{A_i O_i f_{id}}\right)} \text{ for all } \boldsymbol{d}.$$

From that equation it shown that value of $\boldsymbol{A_i^m}$ and $\boldsymbol{B_d^m}$ depend each other and it achieved by iteration process. Give starting point $\boldsymbol{B_d}$ = 1 for all d, so we could get the value of $\boldsymbol{A_i}$. This value of $\boldsymbol{A_i}$ is used for equation (4) to get value of $\boldsymbol{B_d}$. It would be continuously until $\boldsymbol{A_i}$ and $\boldsymbol{B_d}$ value achieve convergence. The first value for balancing factor not influence the final result, otherwise influence to $\boldsymbol{n}$ iteration to get convergence.

For the simplification purposes, we define the following terms as follows:

$[\boldsymbol{T_{id}}]$	=	the **observed O-D matrix** from origin ***i*** to destination ***d***.
$\boldsymbol{O_i^m}$	=	the **total trips** of each mode ***m*** generated by origin ***i***.
$\boldsymbol{D_d^m}$	=	the **total trips** of each mode ***m*** attracted by destination ***d***.
$\boldsymbol{A_i^m}, \boldsymbol{B_d^m}$	=	the **balancing factors** for each mode ***m*** for origin ***i*** and destination ***d***.
$\boldsymbol{C_{id}^m}$	=	the **trip cost of travelling** from origin ***i*** to destination ***d*** by mode ***m***.
$\boldsymbol{\beta}$	=	the **unknown estimated parameter** to be calibrated.
$\boldsymbol{m}$	=	the total **number of modes**.

N	=	the total **number of origins** or **destinations**.
f_{id}	=	a calibration term for interchange ***id***, (friction factor) or travel time factor $f_{id} = e^{-\beta C_{id}}$
i	=	origin zone
d	=	destination zone

We use the notational conventional that $\sum_m$ means the summation begins at ***m***=**1** and continues over the entire range of the subscript.

4. METHODOLOGY

Newton–Raphson method is an efficient algorithm for finding approximations to the zeros (or roots) of a real-valued function.Using Newton-Raphson Method, we should find derivative function of Ai and Bd. It produces iteratively a sequence of approximations to the root, their rate of convergence to the root is quadratic. It can also be used to find a minimum or maximum of such a function, by finding a zero in the function's first derivative. (Weisstein and Eric W.,2008). For further explanation, is shown as bellow.

$$\frac{\partial A_i}{\partial \beta} = -(A_i)^2 \left[\sum_d \left\{ D_d \left(B_d \frac{\partial f_{id}}{\partial \beta} + f_{id} \frac{\partial B_d}{\partial \beta} \right) \right\} \right] \qquad \frac{\partial B_d}{\partial \beta} = -(B_d)^2 \left[\sum_i \left\{ O_i \left(A_i \frac{\partial f_{id}}{\partial \beta} + f_{id} \frac{\partial A_i}{\partial \beta} \right) \right\} \right]$$

$$\frac{\partial^2 A_i}{\partial \beta^2} = 2(A_i)^3 \left[\sum_d \left\{ D_d \left(B_d \frac{\partial f_{id}}{\partial \beta} + f_{id} \frac{\partial B_d}{\partial \beta} \right) \right\} \right]^2 - (A_i)^2 \left[\sum_d \left\{ D_d \left(B_d \frac{\partial^2 f_{id}}{\partial \beta^2} + 2 \frac{\partial B_d}{\partial \beta} \frac{\partial f_{id}}{\partial \beta} + f_{id} \frac{\partial^2 B_d}{\partial \beta^2} \right) \right\} \right]$$

$$\frac{\partial^2 B_d}{\partial \beta^2} = 2(B_d)^3 \left[\sum_i \left\{ O_i \left(A_i \frac{\partial f_{id}}{\partial \beta} + f_{id} \frac{\partial A_i}{\partial \beta} \right) \right\} \right]^2 - (B_d)^2 \left[\sum_i \left\{ O_i \left(A_i \frac{\partial^2 f_{id}}{\partial \beta^2} + 2 \frac{\partial A_i}{\partial \beta} \frac{\partial f_{id}}{\partial \beta} + f_{id} \frac{\partial^2 A_i}{\partial \beta^2} \right) \right\} \right]$$

Before the gravity model can be used for prediction of future travel demand, it must be calibrated. Calibration is accomplished by adjusting the various factors within the gravity model until the model can duplicate a known base year’s trip distribution. For example, if you knew the trip distribution for the current year, you would adjust the gravity model so that it resulted in the same trip distribution as was measured for the current year. The method to calibrate the balancing factor in gravity model is shown as bellow.

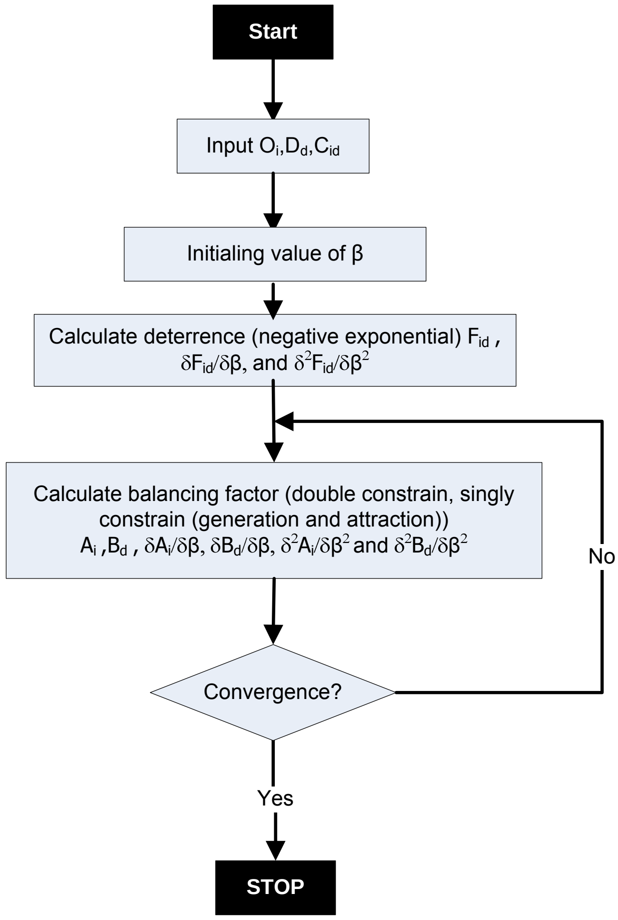

Figure 2 Method of calibration balancing factor in gravity model

5. RESULT

This section present some result of the pattern of balancing factor in Gravity to achieve convergence with any beta values as input . We put beta value as 0.01, 0.05, 0.1, 0.5, 1, 2 and 3. Using Matlab program, we calculate the balancing factor ***Ai*** and ***Bd*****,** first derivative $\boldsymbol{\partial Ai/\partial \beta}$ and $\boldsymbol{\partial Bd/\partial \beta}$ and calculate second derivative $\boldsymbol{\partial^2 Ai/\partial \beta^2}$ and $\boldsymbol{\partial^2 Bd/\partial \beta^2}$.

a. Ai

Table 1 Iteration Beta Value as Input for Calculate Ai

Iteration	Beta 0.01	Beta 0.05	Beta 0.1	Beta 0.5	Beta 1	Beta 2	Beta 3
1	0.00156	0.008545	0.026168	0.077422	0.105898	0.107424	0.107429
2	3.83E-06	0.000458	0.00271	0.413982	0.628105	0.638962	0.638995
3	5.03E-09	2.14E-05	0.000648	0.685995	1.311641	1.342932	1.343014
4	7.53E-12	9.13E-07	0.000117	0.854713	2.292533	2.370142	2.370314
5	1.15E-14	3.86E-08	1.98E-05	0.891637	3.757836	3.947576	3.947921
6	8.88E-16	1.64E-09	3.3E-06	0.821214	5.953823	6.4296	6.430296
7	8.88E-16	6.93E-11	5.47E-07	0.701899	9.151556	10.37844	10.37987
…..							
……							
9029	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	6.1E-05
9030	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	3.05E-05
9031	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	3.05E-05
9032	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	0
9033	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	0

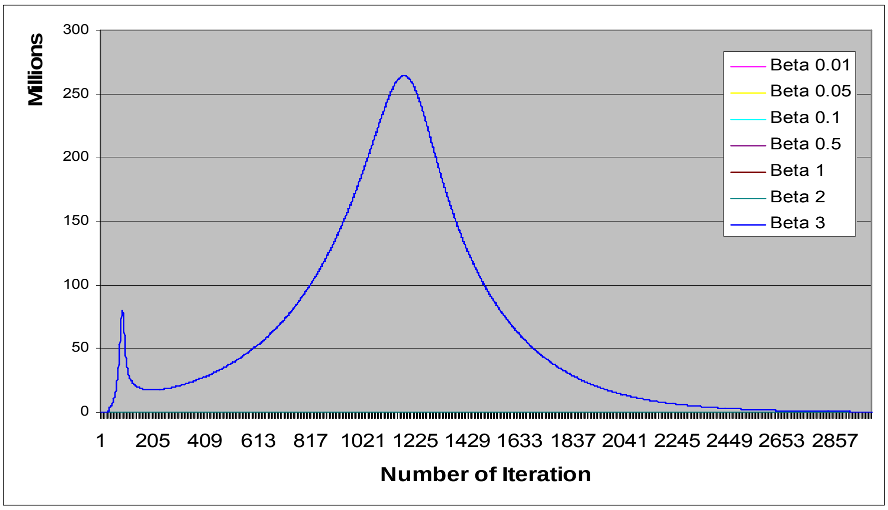

Figure 3 The Number of Iteration with Several Beta Value as Input for Ai

As shown in Table 1 and Figure 3, we know that with value of beta=0.01, we achieve convergence level in iterative 8th. If we put value of beta=0.05, we need iterative 11th to achieve convergence. It have same pattern until iterative 9032 to achieve convergence with the value of beta=3. So, the bigger we put value for beta, the longer iterative to achieve convergence.

b. ***Bd***

Table 2 Iteration Beta Value as Input for Calculate Bd

Iteration	Beta 0.01	Beta 0.05	Beta 0.1	Beta 0.5	Beta 1	Beta 2	Beta 3
1	0.001778	0.002465	0.003494	0.009568	0.010402	0.010443	0.010443
2	6.11E-08	2.68E-06	1.28E-05	0.000653	0.001232	0.001283	0.001283
3	1.2E-11	1.51E-08	1.55E-07	0.00086	0.002132	0.002284	0.002285
4	1.48E-15	2.77E-10	7.78E-08	0.000897	0.003228	0.003628	0.00363
5	7.59E-18	2.07E-11	1.87E-08	0.000818	0.00457	0.005548	0.005552
6	4.34E-19	9.64E-13	3.42E-09	0.000684	0.006132	0.008374	0.008386
7	0	4.17E-14	5.83E-10	0.00055	0.007755	0.012596	0.012625
…..							
……							
9260	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	1.79E-07
9261	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	1.19E-07
9262	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	5.96E-08
9263	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A
9264	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A

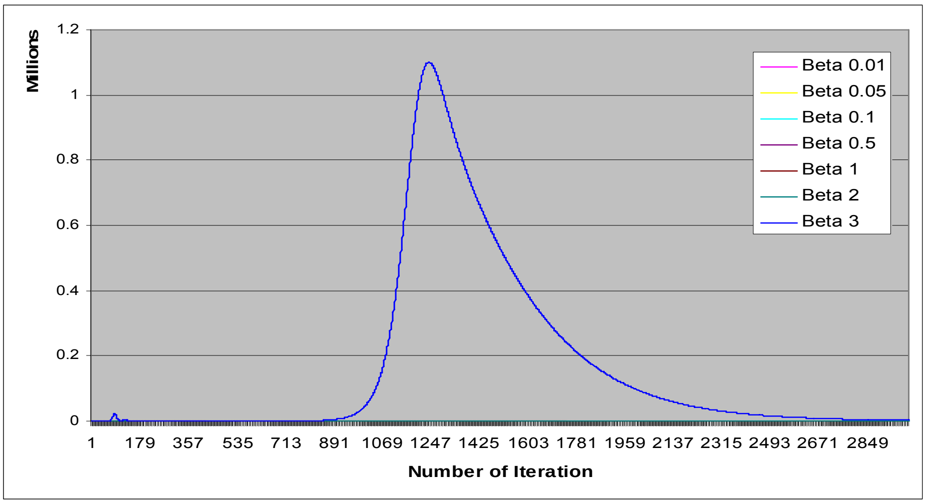

Figure 4 The Number of Iteration with Several Beta Value as Input for Bd

As shown in Table 2 and Figure 4, we know that with value of beta=0.01, we achieve convergence level in iterative 7th. If we put value of beta=0.05, we need iterative 12th to achieve convergence. It have same pattern until iterative 9263 to achieve convergence with the value of beta=3. So, the bigger we put value for beta, the longer iterative to achieve convergence.

c. $\partial Ai/\partial \beta$

Table 3 Iteration Beta Value as Input for Calculate $\partial Ai/\partial \beta$

Iteration	Beta 0.01	Beta 0.05	Beta 0.1	Beta 0.5	Beta 1	Beta 2	Beta 3
1	1.294812	0.821618	1.488994	24975.68	3.27E+10	1.25E+23	1.56E+35
2	0.002818	0.11148	0.759096	16515.92	2.21E+10	8.67E+22	1.11E+35
3	3.88E-06	0.005211	0.15831	11770.87	1.65E+10	6.68E+22	8.85E+34
4	5.91E-09	0.000223	0.027798	8951.065	1.34E+10	5.59E+22	7.67E+34
5	9.31E-12	9.44E-06	0.004681	7162.507	1.15E+10	4.95E+22	6.97E+34
6	#N/A	4E-07	0.000778	5949.906	1.04E+10	4.54E+22	6.52E+34
7	#N/A	1.69E-08	0.000129	5075.446	9.69E+09	4.27E+22	6.2E+34
…..							
……							
8253	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	2.36E+21
8254	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	4.72E+21
8255	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	4.72E+21
8256	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A
8257	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A

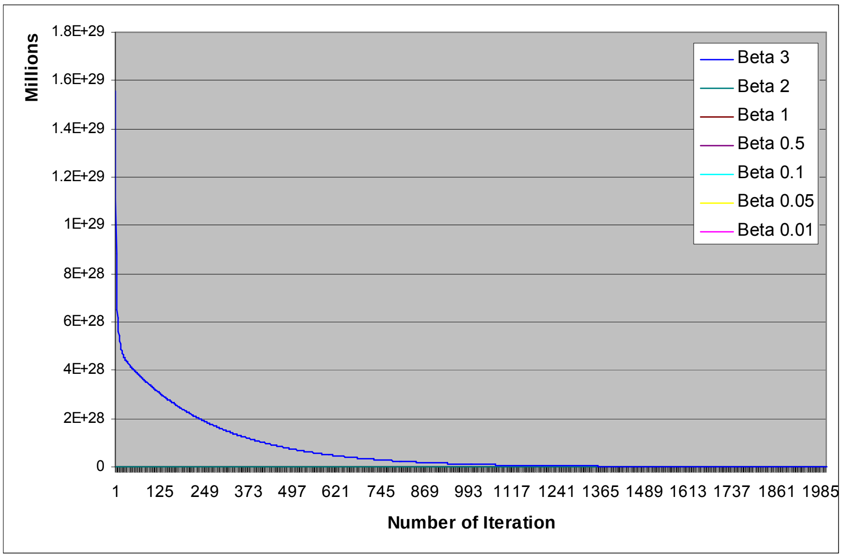

Figure 5 The Number of Iteration with Several Beta Value as Input for *∂Ai/∂β*

As shown in Table 3 and Figure 5, we know that with value of beta=0.01, we achieve convergence level in iterative 6^{th}. If we put value of beta=0.05, we need iterative 24^{th} to achieve convergence. It have same pattern until iterative 8256 to achieve convergence with the value of beta=3. So, the bigger we put value for beta, the longer iterative to achieve convergence.

d. *∂Bd/∂β*

Table 4 Iteration Beta Value as Input for Calculate ***∂Bd/∂β***

Iteration	Beta 0.01	Beta 0.05	Beta 0.1	Beta 0.5	Beta 1	Beta 2	Beta 3
1	0.093246	0.130724	0.190485	21.36944	35143278	3.05E+20	7.02E+32
2	3.69E-05	0.000513	0.001559	16.68217	34136471	2.98E+20	6.87E+32
3	6.72E-09	2.41E-06	3.85E-05	14.13229	33224361	2.91E+20	6.73E+32
4	1.89E-12	8.03E-08	2.25E-05	12.27604	32379524	2.85E+20	6.59E+32
5	6.23E-15	5.18E-09	4.63E-06	10.85308	31585256	2.79E+20	6.46E+32
6	#N/A	2.37E-10	8.17E-07	9.713661	30830928	2.73E+20	6.33E+32
7	#N/A	1.02E-11	1.38E-07	8.768866	30109378	2.67E+20	6.2E+32
…..							
……							
7987	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	1.84E+19
7988	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	1.84E+19
7989	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	1.84E+19
7990	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	0
7991	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	0

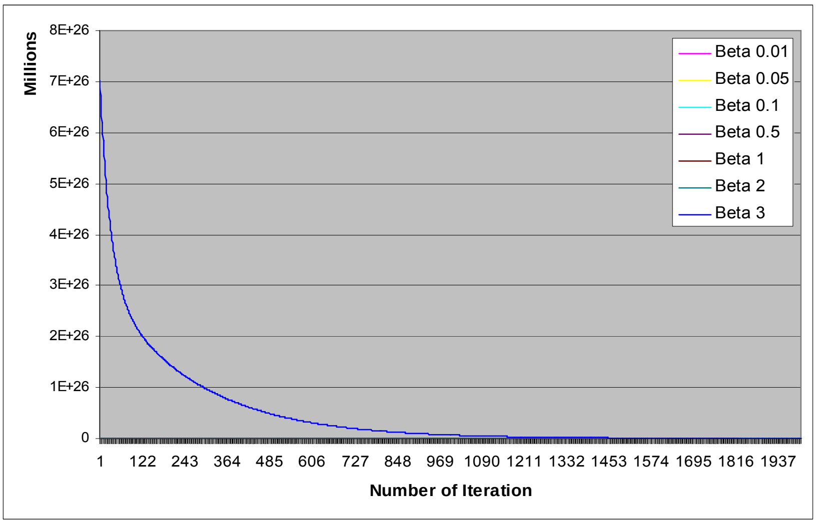

Figure 6 The Number of Iteration with Several Beta Value as Input for $\partial Bd/\partial\beta$

As shown in Table 4 and Figure 6, we know that with value of beta=0.01, we achieve convergence level in iterative 6^{th}. If we put value of beta=0.05, we need iterative 23^{rd} to achieve convergence. It have same pattern until iterative 7990 to achieve convergence with the value of beta=3. So, the bigger we put value for beta, the longer iterative to achieve convergence.

e. $\partial^2 Ai/\partial \beta^2$

Table 5 Iteration Beta Value as Input for Calculate $\partial^2 Ai/\partial \beta^2$

Iteration	Beta 0.01	Beta 0.05	Beta 0.1	Beta 0.5	Beta 1	Beta 2	Beta 3
1	160.248	219.6389	986.6004	1.18E+09	8.86E+19	1.59E+41	6.09E+61
2	0.250657	17.8852	239.6214	7.53E+08	5.7E+19	1.1E+41	4.34E+61
3	0.00036	0.801263	42.84941	5.22E+08	4E+19	8.4E+40	3.47E+61
4	5.56E-07	0.034184	7.244364	3.88E+08	3.06E+19	6.98E+40	3.01E+61
5	8.77E-10	0.001449	1.204915	3.05E+08	2.51E+19	6.13E+40	2.74E+61
6	6.82E-13	6.14E-05	0.199319	2.5E+08	2.17E+19	5.6E+40	2.56E+61
7	1.71E-12	2.6E-06	0.032909	2.11E+08	1.95E+19	5.24E+40	2.43E+61
…..							
……							
8718	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	1.46E+48
8719	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	1.46E+48
8720	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	1.46E+48
8721	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A
8722	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A

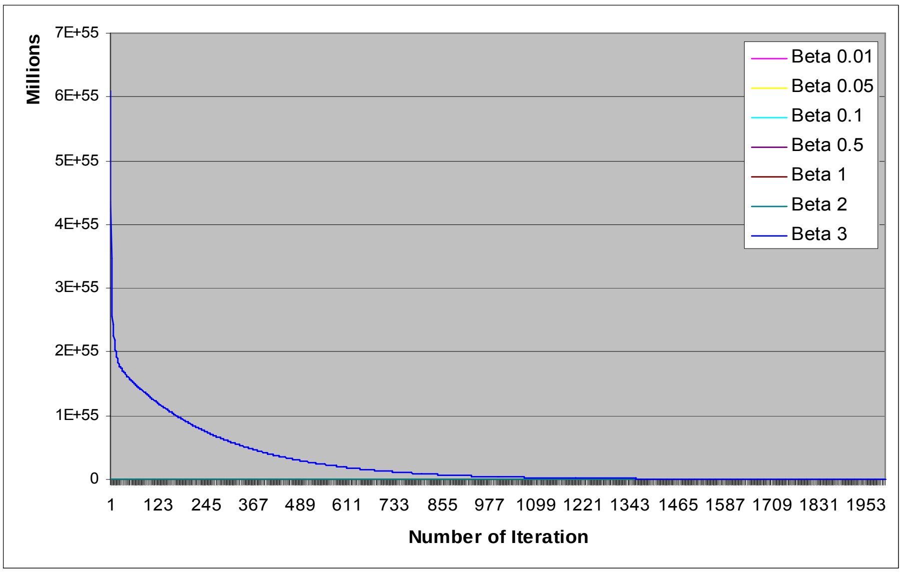

Figure 7 The Number of Iteration with Several Beta Value as Input for $\partial^2 \mathbf{Ai}/\partial\, \boldsymbol{\beta}^2$

As shown in Table 5 and Figure 7, we know that with value of beta=0.01, we achieve convergence level in iterative 158th. If we put value of beta=0.05, we need iterative 158th to achieve convergence. It have same pattern until iterative 8721 to achieve convergence with the value of beta=3. So, the bigger we put value for beta, the longer iterative to achieve convergence.

f. $\partial^2 Bd/\partial\beta^2$

Table 6 Iteration Beta Value as Input for Calculate $\partial^2 Bd/\partial\beta^2$

Iteration	Beta 0.01	Beta 0.05	Beta 0.1	Beta 0.5	Beta 1	Beta 2	Beta 3
1	5.110642	7.724222	13.54406	10788470	4.55E+18	2.86E+40	3.56E+61
2	0.002635	0.034941	0.024567	649748.6	2.26E+16	1.36E+38	1.02E+59
3	4.3E-07	4.18E-05	0.032973	546568.7	2.2E+16	1.35E+38	1.02E+59
4	2.33E-10	1.64E-05	0.007071	472453	2.15E+16	1.34E+38	1.02E+59
5	6.03E-13	8.35E-07	0.00126	416369.7	2.11E+16	1.33E+38	1.02E+59
6	8.88E-16	3.68E-08	0.000213	371933.2	2.07E+16	1.32E+38	1.02E+59
7	8.88E-16	1.57E-09	3.55E-05	335368.9	2.04E+16	1.31E+38	1.02E+59
…..							
……							
7795	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	2.28E+46
7796	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	2.28E+46
7797	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	2.28E+46
7798	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	0
7799	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	0

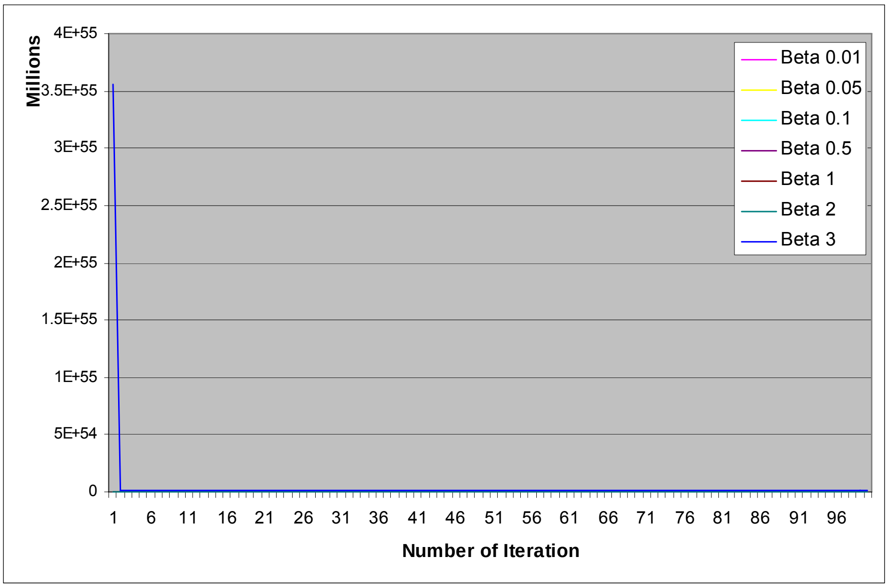

Figure 8 The Number of Iteration with Several Beta Value as Input for $\partial^2 Ai/\partial\beta^2$

As shown in Table 6 and Figure 8, we know that with value of beta=0.01, we achieve convergence level in iterative 157th. If we put value of beta=0.05, we need iterative 157th to achieve convergence. It have same pattern until iterative 7798 to achieve convergence with the value of beta=3. So, the bigger we put value for beta, the longer iterative to achieve convergence.

The number of iteration for each beta value to achieve convergence in ***Ai, Bd,*** $\partial Ai/\partial\beta$, $\partial Bd/\partial\beta$, $\partial^2 Ai/\partial\beta^2$ and $\partial^2 Bd/\partial\beta^2$ is described in **Table 7**.

Table 7 The Number of Iteration *of* ***Ai, Bd,*** $\partial Ai/\partial\beta$, $\partial Bd/\partial\beta$, $\partial^2 Ai/\partial\beta^2$ and $\partial^2 Bd/\partial\beta^2$

	Number of Iteration (Iteration *n*)						
	β =0.01	**β =0.05**	**β =0.1**	**β =0.5**	**β =1**	**β =2**	**β = 3**
Ai	8	11	21	405	1675	4272	9032
Bd	7	12	20	421	1746	4284	9263
∂Ai/∂β	6	24	24	1824	1824	3788	8256
∂Bd/∂β	6	23	23	1824	1824	3550	7990
$\partial^2 Ai/\partial\beta^2$	158	158	158	506	1414	3965	8721
$\partial^2 Bd/\partial\beta^2$	157	157	157	505	1426	3512	7798

Table 7 shown that to get convergence in Ai value, if we put beta value=0.01 (***β*=0.01**), ***Ai*** would convergent in 8th iteration and so with other value of beta. Second derivative of Ai and Bd need more iteration than others.

6. CONCLUSION

The pattern of balancing factor in Gravity to achieve convergence with any beta values as input is presented using Matlab program. It is shown, by way of exponential as friction factor and using Double Constraint Gravity as balancing factor, we could see that the bigger beta value as input (***β*>1**), the longer iteration to achieve convergence.

REFERENCES

Allen, B. (1984), *Trip Distribution Using Composite Impedance,* Transportation Research

Record 944 pp. 118-127

Evans, S.P., (1976), *Derivation and Analysis of Some Models for Combining Trip Distribution and Assignment*. Transportation Research 10, 37–57.

Levinson, D. and Kumar A. (1995). *A Multi-modal Trip Distribution Model. Transportation Research Record* #1466: 124-131.

Ortuzar, J.D. and Willumsen, L.G. (1994), ***Modelling Transport*, Second Edition**, John Wiley and Sons Ltd.

Suyuti, R. dan Tamin, O.Z. (2006), ***Estimasi Parameter Model Kebutuhan Transportasi Berdasarkan Informasi Data Arus Lalu Lintas Pada Kondisi Pemilihan Rute Keseimbangan*n,** Desertasi Doktor, Institut Teknologi Bandung

Tamin, O.Z. (1985), ***The Estimation of Matrices for Freight Movement From Traffic Counts Using a Non-Linear Regression Approach***, MSc Thesis of the University of London, Imperial College and University College London.

Tamin, O.Z. (1988), ***The Estimation of Transport Demand Model From Traffic Counts***, PhD Dissertation of the University of London, University College London.

Tamin, O.Z. and Willumsen, L.G. (1989), *Transport Demand Model Estimation From Traffic Counts*, **Journal of Transportation**, 16, 3-26.

Tamin, O.Z. (1992). *Application of Transport Demand Models for Inter-Regional Vehilce Movements in East Java (Indonesia)*. Proceedings of the 6th World Conference on Transport Research (WCTR), Lyon, France.

Tamin, O.Z. (2000), ***Transport Planning and Modelling*, 2nd Edition**. ITB Press, Bandung.

Tamin, O.Z., et al (2001), *Dynamic Origin-Destination (O-D) Matrices Estimation From Real Time Traffic Count Information,* Graduate Team Research, University Research for Graduate Education (URGE) Project

Tamin, O.Z., Sjafruddin A, and Purwanti, O. (2002), *Public Transport Demand Estimation by Calibrating The Combined Trip Distribution-Mode Choice (TDMC) Model from Passenger Counts: A Case Study in Bandung, Indonesia*, the 4th Proceedings of the Eastern Asia Society for Transportation Studies (EASTS).