Memoirs of the Faculty of Engineering, Okayama University, Vol.23, No. 2, March 1989

# *On Robust Incomplete Choleski-Conjugate Gradient Method And Its Modification*

Takeo Taniguchi[+] & Kohji Fujiwara[++]

(Received January 24, 1989)

SYNOPSIS

This paper includes a solver for a large sparse set of linear algebraic equations which are obtained by the application of the finite element method to static structural problems. Proposed method is a modification of Robust Incomplete Choleski-Conjugate Gradient Method, which belongs to Preconditioned Conjugate Gradient Method suitable for supercomputers. Through a number of numerical experiments the authors show that Robust Incomplete Choleski-Conjugate Gradient Method sometimes fails in to obtain the solutions, secondly they clarify the reason of the failures from the theoretical viewpoint, and finally they propose a modification of the robust method by the introduction of the theoretical result. Proposed method is as effective as the original, and it can overcome the demerit of Robust Method which is clarified through numerical experiments.

## 1. INTRODUCTION

The recent development of the supercomputer requires newer and faster solvers for large sparse sets of linear algebraic equations. The characteristics of the computers require us more effective utilization of CPU memory for large-scale problems, and therefore, new solvers are necessarily selected among iterative ones. Thus, solvers belonging to

+ : Engineering Science Department ++: Civil Engineering Department

conjugate gradient method are thought to be one of the best for this purpose.[1] Though the method generally shows good convergence behaviour, it is still insufficient for large-scale problems which engineers encounter, and the improvement of the convergence behaviour has been required. One of the best ways for this improvement is the addition of the preconditioner for CG method, and at present we find a number of preconditioners.[2]

The role of a preconditioner is the improvement of the distribution of all eigenvalues of a set of linear algebraic equations before the application of CG method. All of the preconditioners at present in use are classified into two types; the matrix splitting and matrix factorization methods. Several preconditioners belonging to the latter are already proposed, and the most popular one is the incomplete Choleski factorization method.[2] Henceforce, we call the conjugate gradient method with a preconditioner as PCG method.

PCG method is widely used in engineering fields, and in most cases it gives good results comparing to the direct methods with respect to the execution-time and necessary memories. But, in some cases of structural problems PCG method cann't be effectively applied or fails in the computations. For overcoming this failure artificial methods are also proposed. One of them is the robust incomplete Choleski-conjugate gradient method which was proposed by Ajiz and Jennings, and a number of investigations clarified its efficiency of the application in many engineering problems.[3]

The main purpose of this investigation is to survey the efficiency of the robust method. Through a number of numerical experiments of the method we show that in some cases the method fails in the computations of the matrix factorization. Then, we clarify the reason of the failure. and by considering the reason we propose modified robust incomplete Choleski-conjugate gradient method.

## 2. PRECONDITIONED CONJUGATE GRADIENT METHOD

Let

$$Ax = b \tag{1}$$

be a set of linear algebraic equations. Our aim is to solve the unknown vector x by using PCG method.

Let B be a matrix which is appropriately determined, and the multiplication of B to (1) yields to another set of linear algebraic equation

$$BAx = Bb \tag{2}$$

If the matrix BA is equal to a unit matirx (i.e. B is the inverse matrix of A), (2) directly gives the solution vector, x.

Let B be a matrix which is similar to the inverse matrix of A. Since the matrix BA is near unit matrix, then the application of CG method to (2) easily leads to the solution comparing to the direct application of CG method to (1).

The procedure mentioned above is the basic idea of PCG method, and the matrix B is called a preconditioner. From these explanation we can recognize the role of a preconditioner as following; the preconditioner works to arrange the distribution of all eigenvalues so that CG method can easily converge to the solution, and, therefore, the preconditioner need not to be a strict inverse matrix of A but may be similar to it.

There are a number of methods to find and construct B matrix.[2] These methods are classified into two categories; matrix splitting and matrix factorization methods. Among these methods the former is usually introduced in engineering fields, and the most popular one is the incoplete Choleski factorization method, which is a kind of Choleski factorization method and factorizes off-diagonal elements in accordance with some specified criterions. Henceforce, we call CG method with incomplete Choleski factorization method as ICCG. Now, we explain the incomplete Choleski factorization.

Before the explanation, we briefly summarize the process of Choleski factorization, since the process has the important meaning at the reasoning of the failures of robust incomplete Choleski-conjugate gradient method. The process of Choleski factorization is expressed as following; For k = 2,3, ..., n,

$$a_{kk} = \left( a_{kk} - \sum_{j=1}^{k-1} a_{kj}^{2} \right)^{1/2} \tag{3}$$

$$a_{ik} = \left( a_{ik} - \sum_{j=1}^{k-1} a_{ij} a_{kj} \right) / a_{kk} \tag{4}$$

, where $k < n$ and $i = k+1, k+2, \ldots, n$.

In above expressions we should notice that since the main diagonal entries are subjected to the root computation, they must be always positive through the factorization process. In other word, if there arises a negative value at the diagonal entry, the computation stops at the stage.

Now, we explain the incomplete Choleski factorization. At first, determine the set of elements P which are not factorized at Choleski decomposition as

$$P = \{ (i,j) \mid L_{ij} = 0,\ 1 < i,j < n \} \tag{5}$$

Successively, we continue the Choleski factorization so that

$$\left.\begin{array}{l} \text{if } (i,j) \in P, \text{ then } L_{ij} = 0, \\ \text{if } (i,j) \notin P, \text{ then } Q_{ij} = A_{ij} \end{array}\right\} \tag{6}$$

At the selection of the set P there are a number of methods, and thus, we can give several ICCG methods. They generally work very well for many kind of problems, but sometimes we fail in to obtain solution for following problems which include plate, shell and isoparametric finite elements. This failure is due to the appearance of negative value at the main diagonal entry. The reason of the occurrence of negative value is obviously due to the incompleteness of the factorization process for the coefficient matrix, and in order to prevent the occurrence of the failures during the factorization, Ajiz and Jennings proposed so-called robust incomplete Choleski-conjugate gradient method .[3] Since by (4) the diagonal entry is modified by off-diagonal values , it may happen that the diagonal value becomes negative by the procedure neglecting the factorization of off-diagonals. Then, this method adds to the diagonal entry a slight value which depends on the neglected values at the factorization process in order to prevent the appearance of negative value.

Here, we explain the robust method by Ajiz and Jennings briefly. Let $a_{ij}^*$ be an off-diagonal element of the coefficient matrix in the factorization process, and $\psi$ be a parameter which is arbitrarily determined as $0 < \psi < 1$ by the user. If any off-diagonal element satisfies

$$a_{ij}^{*2} < \psi^2 a_{ii} a_{jj} \tag{7}$$

, then the element is not factorized, and its diagonal element is modified by using following equations;

$$a_{ii} = a_{ii} + \sqrt{a_{ij}^{2} a_{jj}/aii} \tag{8}$$

$$a_{jj} = a_{jj} + \sqrt{a_{ij}{}^2 a_{ii}/a_{jj}} \quad (9)$$

Here, we should notice that (8) is used only when the last off-diagonal entry of the i-th row is neglected. On the other hand, the modification at (9) is always applied for the modification of $a_{jj}$.

From (7) it is obvious that the method is equivalent to the complete Choleski factorization if $\psi$ is equal to 0. On the other hand, if $\psi$ has rather a big value, then the result of the factorization becomes quite different from the one by the complete Choleski. As a result, even though giving big value for $\psi$ can save necessary CPU memory, it results in the increase of the number of iterations which is required for the computation of the conjugate gradient method. This indicates that the setting of a parameter $\psi$ is the most important factor at the use of robust method, because it decides not only the CPU-time but also CPU memory necessary for the method.

If any solver is required to be introduced as a general-purpose method, it is necessarily applied to various types of linear algebraic equations. Of course, it is also required to be fast and reliable. In successive section we survey on these two faces of the robust incomplete Choleski-conjugate gradient method.

## 3. ON THE VALIDITY OF ROBUST INCOMPLETE CHOLESKI-CONJUGATE GRADIENT METHOD

Robust incomplete Choleski-conjugate gradient method (henceforce expressed as RCGM) is proposed to prevent the occurrence of the failure of the incomplete factorization process, and the main purpose of this section is to survey whether the method can completely prevent the failure. Since some papers report that the failures occur when we treat "plate", "shell", and "isoparametric" elements at the finite element analysis, the test problems treated here are one of these, i. e. plate elements.[4]

Examples used for our examination are illustrated in Fig.1. Square plate is divided into N and M elements along two axes. The configuration of elements are triangular. As described already, since the convergency is governed by the distribution of eigenvalues of the coefficient matrix, a number of factors which mainly give the important influence on the distribution are taken into consideration at the

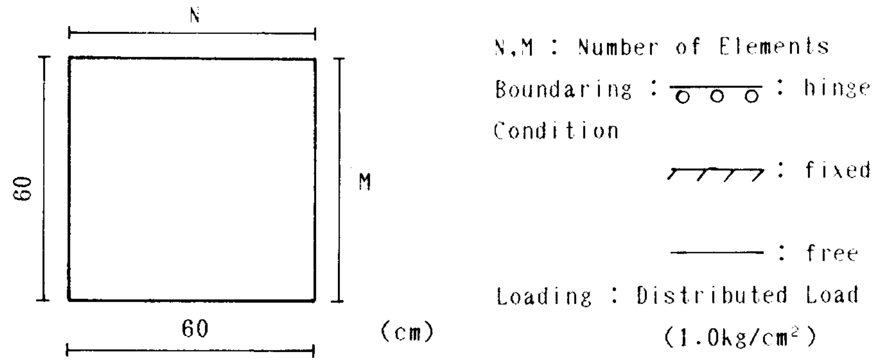


Fig 1 Structual Model and Its Boundary Conditions

finite element modellings, for example the boundary conditions, the mesh sizes, and so on. Since the parameter $\psi$ plays the most important role at the use of RCGM, the results of our numerical experiments are summarized with respect to the values of $\psi$. Note that the convergence condition of the results is set to be $10^{-6}$ for all test problems.

The results are summarized in Tables 1 to 5. In these tables PSI indicates the value of $\psi$, RCGM indicates the ratio of nonzero entries required for incomplete and complete Choleski factorizations, and IT indicates the number of iterations required for the conjugate gradient method. From the results we can notice that the state of the convergency is largely governed by the prescribed factors which determine the distribution of eigenvalues. At the same time, we also find that in some numerical experiments RCGM failed in to obtain solutions. These cases are indicated by using the mark "X" in the tables. Since the test problem is the same one, these failures are caused by the value of $\psi$. Moreover, we cann't find any tendencies of the values $\psi$'s which lead to failures. Only one tendency we can find out is that the occurrence of the failures increases in accordance with the size of problems.

Now, let consider on the process of the occurrence of these failures From the characteristic of the incomplete factorization the beginning is the neglection of a nonzero off-diagonal element. We assume that at least one off-diagonal element of the i-th column, which does not locate at the last position in the row, is neglected at the factorization. Then, (9) is applied for the modification of $a_{jj}$, and $a_{jj}$ is necessarily overestimated by (9) comparing to the value due to the complete Choleski factorization.

We consider on the entries in the i-th row, which are subjected to the numerical operation of (4). The neglection of off-diagonal entries overestimates the value in the bracket of (4), and the value of $a_{kk}$ is already overestimated. Thus, as a result, the value of $a_{ik}$ of (4) may be sometimes overestimated, and sometimes underestimated. As far as the value of $a_{ik}$ is underestimated, there arise no problem at the factorization process, because from (3) the influence of the under-estimation of off-diagonal entries overestimates the value of $a_{kk}$, and negative value never appear at $a_{kk}$. But, if $a_{ik}$ is overestimated and if it does not locate at the last position, it gives serious influence. That is, since $a_{kk}$ must be modified by (7), $a_{kk}$ may become negative and, as a result, the root computation of (3) becomes impossible. This is the reasoning of the occurrence of the numerical failure of the robust method.

## 3. MODIFIED ROBUST INCOMPLETE CHOLESKI-CONJUGATE GRADIENT METHOD

In previous section we could clarified the phenomenon of the numercal failure at the use of the robust method. This section is used for the proposal of the modification of RCGM which can prevent the occurrence of negative values at the main diagonals.

According to RCGM the diagonal elements (namely $a_{ii}$) is necessarily modified by (9) when any off-diagonal element in the i-th column is neglected. But, the modification by (8) due to the neglection of off-diagonal entry is applied only when the last off-diagonal entry is neglected. Thus, if the diagonal entry can be modified by any neglection of off-diagonal element by the application of (8), the diagonal value is always modified positively. On the value of the modification , i.e. $a_{ij}$ of (8), we use the maximum value among the neglected ones in the row. From above consideration we can propose following method which is a modification of RCGM:

[MODIFIED ROBUST INCOMPLETE CHOLESKI-CONJUGATE GRADIENT METHOD]

If some off-diagonal elements $a_{ij}$'s are neglected in the i-th row at the incomplete factrization process by the judgement of (7), then modify $a_{ii}$ by using (8).

Now, let's survey the efficiency of the proposed method. Henceforce, we call the modified robust method as MRCGM. All of the test problems presented in previous section are used for this purpose, and the results of numerical experiments are also summarized in the same tables (see Tables 1 to 5). The results by MRCGM are given in the columns of MRCGM in these tables. The results show that the proposed method can always lead to converged solutions for all cases including those which cann't be obtained by RCGM. On the aspect of the number of iterations to obtain solutions by CG method, we can notice that the modified method (MRCGM) requires slightly longer execution time comparing to RCGM.

## 4. CONCLUDING REMARKS

In this paper the authors surveyed the efficiency of Robust Incomplete Choleski-Conjugate Gradient Method, and they could clarify that 1) the method sometimes fails in to obtain the converged solutions, and 2) the failures are caused by the insufficient modification of main diagonal element. By taking into consideration of this reasoning, they could propose a modification method of the robust method, and a number of numerical experiments showed the efficiency of the modified method.

As the concluding remarks, we can list that 1) proposed method is more effective than the original one as a general-purpose solver for a large sparse set of linear algebraic equations, and 2) it is as effective as the original on the aspect of the execution-time.

## REFERENCES

[1] M.R. Hestenes and E. Stiefel,"Method of conjugate gradients for solving linear systems", J. Res. Nat. Bur. Standards, No.49, pp. 409-436 (1952)

[2] T. Nodera,'PCG method for large-scale sparse matrix', Seminar on mathematical science, Vol.7, Keioh University (1983) (in Japanese)

[3] M.A. Ajiz and A. Jennings,"A robust incomplete Choleski-conjugate gradient algorithm", Int. J. Numer. Meth. Engg, Vol.20, pp.949-966 (1984)

[4] T. Taniguchi,"Large-scale matrix computation in civil engineering field", Supercomputer and Large-scale Computation; bit, Vol.19, No.13, pp.1699-1709 (1987) (in Japanese)

Table 1

| | No.of Variables = 96 | | | | No.of Variables = 199 | | | | No.of Variables = 303 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PSI | RCGM | IT | MRCGM | IT | RCGM | IT | MRCGM | IT | RCGM | IT | MRCGM | IT |
| 0.01 | 0.642 | 13 | 0.631 | 16 | 0.496 | 21 | 0.484 | 22 | 0.399 | 26 | 0.391 | 30 |
| 0.02 | 0.576 | 17 | 0.564 | 21 | 0.452 | 26 | 0.446 | 28 | 0.365 | 37 | 0.350 | 43 |
| 0.03 | 0.540 | 20 | 0.526 | 25 | 0.418 | 29 | 0.411 | 34 | 0.329 | 44 | 0.318 | 50 |
| 0.04 | 0.519 | 24 | 0.507 | 27 | 0.386 | 37 | 0.377 | 42 | 0.285 | 64 | 0.292 | 55 |
| 0.05 | 0.488 | 28 | 0.472 | 32 | 0.363 | 51 | 0.352 | 51 | 0.263 | 75 | 0.271 | 72 |
| 0.06 | × | | 0.444 | 36 | × | | 0.324 | 59 | 0.253 | 80 | 0.253 | 84 |
| 0.07 | 0.432 | 36 | 0.410 | 43 | × | | 0.315 | 62 | 0.246 | 85 | 0.241 | 92 |
| 0.08 | 0.406 | 40 | 0.383 | 47 | × | | 0.301 | 65 | 0.232 | 93 | 0.234 | 94 |
| 0.09 | 0.386 | 42 | 0.369 | 48 | 0.279 | 81 | 0.265 | 86 | 0.227 | 95 | 0.227 | 95 |
| 0.10 | 0.361 | 47 | 0.342 | 52 | 0.255 | 87 | 0.248 | 91 | 0.207 | 110 | 0.210 | 111 |

M/2 = 6
N/2 = 6

M/2 = 8
N/2 = 9

M/2 = 9 (B.C.)
N/2 =12

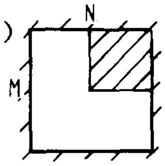


Table 2

| | No.of Variables = 96 | | | | No.of Variables = 199 | | | | No.of Variables = 303 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PSI | RCGM | IT | MRCGM | IT | RCGM | IT | MRCGM | IT | RCGM | IT | MRCGM | IT |
| 0.01 | 0.622 | 17 | 0.608 | 21 | 0.475 | 23 | 0.469 | 28 | × | | 0.382 | 37 |
| 0.02 | 0.552 | 24 | 0.549 | 28 | 0.425 | 32 | 0.416 | 38 | 0.356 | 40 | 0.349 | 47 |
| 0.03 | 0.530 | 27 | 0.522 | 33 | 0.395 | 41 | 0.385 | 45 | 0.322 | 51 | 0.311 | 59 |
| 0.04 | 0.504 | 31 | 0.495 | 38 | 0.376 | 50 | 0.366 | 52 | × | | 0.300 | 65 |
| 0.05 | 0.466 | 36 | 0.453 | 43 | 0.356 | 56 | 0.343 | 62 | × | | 0.277 | 86 |
| 0.06 | 0.443 | 36 | 0.428 | 46 | × | | 0.323 | 72 | × | | 0.255 | 96 |
| 0.07 | 0.429 | 39 | 0.423 | 48 | 0.313 | 78 | 0.309 | 78 | 0.250 | 95 | 0.246 | 104 |
| 0.08 | 0.425 | 45 | 0.402 | 54 | 0.298 | 79 | 0.293 | 85 | 0.236 | 111 | 0.232 | 115 |
| 0.09 | 0.402 | 46 | 0.389 | 57 | 0.286 | 82 | 0.278 | 88 | 0.223 | 118 | 0.220 | 122 |
| 0.10 | 0.387 | 47 | 0.375 | 58 | 0.259 | 98 | 0.250 | 103 | 0.211 | 130 | 0.211 | 134 |

M/2 = 6
N/2 = 5

M/2 = 8
N/2 = 8

M/2 = 9 (B.C.)
N/2 =11

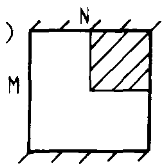

Table 3

| | No.of Variables = 100 | | | | No.of Variables = 203 | | | | No.of Variables = 304 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PSI | RCGM | IT | MRCGM | IT | RCGM | IT | MRCGM | IT | RCGM | IT | MRCGM | IT |
| 0.01 | 0.679 | 33 | 0.637 | 44 | 0.524 | 63 | 0.493 | 75 | × | | 0.389 | 110 |
| 0.02 | 0.591 | 45 | 0.543 | 53 | × | | 0.391 | 93 | 0.321 | 112 | 0.306 | 130 |
| 0.03 | 0.498 | 51 | 0.478 | 60 | 0.359 | 89 | 0.349 | 102 | 0.289 | 142 | 0.266 | 147 |
| 0.04 | 0.461 | 55 | 0.440 | 65 | 0.338 | 102 | 0.316 | 108 | 0.256 | 144 | 0.247 | 157 |
| 0.05 | 0.424 | 62 | 0.403 | 71 | 0.305 | 107 | 0.297 | 121 | 0.235 | 158 | 0.233 | 174 |
| 0.06 | 0.386 | 66 | 0.377 | 77 | 0.285 | 111 | 0.281 | 129 | 0.217 | 168 | 0.212 | 202 |
| 0.07 | 0.358 | 70 | 0.333 | 85 | 0.261 | 123 | 0.250 | 148 | × | | 0.183 | 236 |
| 0.08 | 0.330 | 80 | 0.313 | 91 | 0.220 | 153 | 0.215 | 170 | × | | 0.162 | 272 |
| 0.09 | 0.289 | 86 | 0.286 | 100 | 0.195 | 173 | 0.191 | 187 | × | | 0.141 | 297 |
| 0.10 | 0.268 | 91 | 0.258 | 105 | 0.179 | 192 | 0.180 | 202 | 0.128 | 305 | 0.130 | 311 |

M = 5 N/2 = 6 | M = 7 N/2 = 9 | M = 8 (B.C.) N/2 =12

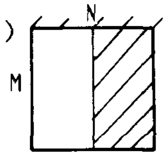


Table 4

| | No.of Variables = 101 | | | | No.of Variables = 196 | | | | No.of Variables = 308 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PSI | RCGM | IT | MRCGM | IT | RCGM | IT | MRCGM | IT | RCGM | IT | MRCGM | IT |
| 0.01 | 0.647 | 15 | 0.633 | 18 | × | | 0.442 | 30 | 0.378 | 33 | 0.374 | 37 |
| 0.02 | 0.579 | 20 | 0.577 | 23 | 0.393 | 32 | 0.380 | 37 | 0.336 | 41 | 0.327 | 47 |
| 0.03 | 0.555 | 23 | 0.550 | 27 | 0.339 | 48 | 0.337 | 49 | 0.312 | 44 | 0.301 | 57 |
| 0.04 | 0.525 | 28 | 0.515 | 32 | 0.301 | 57 | 0.299 | 59 | 0.290 | 64 | 0.282 | 68 |
| 0.05 | 0.487 | 32 | 0.474 | 37 | 0.276 | 63 | 0.274 | 65 | × | | 0.251 | 88 |
| 0.06 | 0.463 | 34 | 0.450 | 40 | 0.267 | 68 | 0.265 | 69 | 0.242 | 86 | 0.239 | 91 |
| 0.07 | 0.444 | 35 | 0.437 | 41 | 0.256 | 71 | 0.254 | 73 | 0.231 | 90 | 0.226 | 97 |
| 0.08 | 0.432 | 40 | 0.415 | 46 | 0.243 | 73 | 0.242 | 77 | 0.219 | 97 | 0.218 | 103 |
| 0.09 | 0.415 | 40 | 0.398 | 49 | 0.234 | 77 | 0.232 | 84 | 0.210 | 101 | 0.203 | 113 |
| 0.10 | 0.396 | 41 | 0.384 | 50 | 0.213 | 86 | 0.221 | 89 | 0.199 | 107 | 0.194 | 118 |

M/2 = 6 N/2 = 5 | M/2 = 6 N/2 =10 | M/2 = 8 (B.C.) N/2 =12

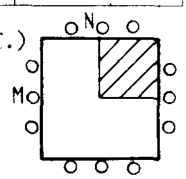

Table 5

| | No.of Variables = 102 | | | | No.of Variables = 202 | | | | No.of Variables = 299 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PSI | RCGM | IT | MRCGM | IT | RCGM | IT | MRCGM | IT | RCGM | IT | MRCGM | IT |
| 0.01 | 0.670 | 21 | 0.657 | 25 | × | | 0.436 | 38 | 0.427 | 33 | 0.418 | 41 |
| 0.02 | 0.619 | 26 | 0.601 | 32 | 0.388 | 39 | 0.376 | 46 | 0.357 | 46 | 0.356 | 54 |
| 0.03 | 0.573 | 31 | 0.547 | 38 | 0.336 | 57 | 0.334 | 61 | 0.350 | 58 | 0.340 | 59 |
| 0.04 | 0.532 | 33 | 0.513 | 43 | 0.301 | 67 | 0.296 | 73 | × | | 0.328 | 67 |
| 0.05 | 0.510 | 35 | 0.502 | 47 | 0.274 | 75 | 0.270 | 79 | 0.310 | 78 | 0.310 | 78 |
| 0.06 | 0.498 | 38 | 0.488 | 51 | 0.265 | 79 | 0.262 | 83 | 0.305 | 81 | 0.294 | 92 |
| 0.07 | 0.472 | 43 | 0.456 | 62 | 0.253 | 86 | 0.251 | 89 | 0.283 | 103 | 0.281 | 103 |
| 0.08 | 0.452 | 49 | 0.434 | 69 | 0.239 | 90 | 0.237 | 96 | 0.272 | 106 | 0.265 | 114 |
| 0.09 | 0.435 | 56 | 0.419 | 72 | 0.230 | 96 | 0.226 | 105 | 0.258 | 121 | 0.252 | 126 |
| 0.10 | 0.411 | 61 | 0.403 | 75 | 0.209 | 105 | 0.216 | 110 | × | | 0.230 | 143 |

M/2 = 7
N/2 = 4

M/2 = 6
N/2 =10

M/2 =10 (B.C.)
N/2 = 9