

ARTICLE

Computing Acceleration for a Pin-by-Pin Core Analysis Method Using a Three-Dimensional Direct Response Matrix Method

Takeshi MITSUYASU*, Kazuya ISHII, Tetsushi HINO and Motoo AOYAMA

Energy and Environmental Systems Laboratory, Hitachi, Ltd.,
7-2-1 Omika-cho Hitachi-shi Ibaraki, 319-1221 Japan

The computing algorithms and acceleration method for the three-dimensional response matrix method were developed. A new formalization for reconstruction of a response matrix is defined by using an inverse matrix and sub-response matrices. The formalization makes possible to define matrix symmetry. This symmetry reduces the number of calculation by half. The developed computing algorithms are the three-dimensional red-black method and the automatic load balancing method. These computing algorithms reduce the computational time. In the ABWR quarter core power and void coupled calculation, the calculation finishes in a practical time with a practical number of PCs.

KEYWORDS: response matrix, neutron current, Monte Carlo, parallelization

I. Introduction

Advanced core designs need precise neutronic calculation modeling, particularly more precise representations of intra- and inter-assembly heterogeneity effects. Monte Carlo calculations are one of the most suitable methods for representing these effects because precise geometrical treatment is possible and no homogenization processes are needed in the calculations. However, Monte Carlo calculations need a long computational time to reduce statistical uncertainty. A direct response matrix method using Monte Carlo calculations has been developed to get the benefit of Monte Carlo calculations without long computational time.¹⁾ The method was expanded to a three-dimensional direct response matrix (3D-DRM) method.^{2,3)} The 3D-DRM method divides neutron behaviors in fuel assemblies into four fundamental reactions and makes four kinds of sub-response matrices including the pin-by-pin fission reactions. These sub-response matrices can be obtained by Monte Carlo calculations in an infinite lattice. A nodal response matrix is reconstructed by these sub-response matrices in each node and the k-effective. Since the out-going partial neutron current is decided by the in-coming partial neutron current and response matrices, the core k-effective is obtained by iterating the reconstruction of response matrices and balancing the neutron currents. Then, as the Monte Carlo calculation is needed only for the lattice calculation in the 3D-DRM method, the computational time for the core analysis becomes practical.

From the viewpoint of computer hardware, computational speed is increasing every year. Increasing speed has been advantageously applied in many fields of computational physics. In high performance computing which uses a lot of

computer resources, it is especially necessary to calculate in parallel.

In this paper, a 3D-DRM method using a Monte Carlo calculation is described along with its parallelization and the computing algorithms developed for parallel computing.

II. 3D-DRM Method and Computing Algorithm

1. Formalization of 3D-DRM

In the 3D-DRM method, the four sub-response matrices are defined as the following.

- (1) Transmission probability: (T)
- (2) Neighbor-induced production probability: (S)
- (3) Self-induced production probability: (A)
- (4) Escape probability: (L)

These sub-response matrices are calculated in the infinite lattice using a Monte Carlo calculation. Since the Monte Carlo calculation system is small, the statistical uncertainty can be reduced within a short computational time. With these sub-response matrices, a k-dependent response matrix, $R(k)$ is represented as

$$R(k)_{ii,gi \rightarrow io,go} = T_{ii,gi \rightarrow io,go} + \frac{1}{k} \sum_{js=1}^{jsmax} \sum_{l=1}^{lmax} S_{ii,gi \rightarrow js,l} [L_{js,l \rightarrow io,go} + \frac{1}{k} \sum_{js2=1}^{jsmax} \sum_{l2=1}^{lmax} A_{js1,l1 \rightarrow js2,l2} [L_{js2,l2 \rightarrow io,go} + \frac{1}{k} \sum_{js3=1}^{jsmax} \sum_{l3=1}^{lmax} A_{js2,l2 \rightarrow js3,l3} [L_{js3,l3 \rightarrow io,go} + \dots]]] \quad (1)$$

where $jsmax$ denotes the number of fuel rods in the corresponding bundle and $lmax$ denotes the number of axial zones in the corresponding fuel rod. $R(k)_{ii,gi \rightarrow io,go}$ is an expected number of neutrons which are induced by a neutron entering from the ii 'th face at the gi 'th group and which eventually exit to the io 'th face at the go 'th group. The face which neu-

*Corresponding author, E-mail: takeshi.mitsuyasu.mz@hitachi.com

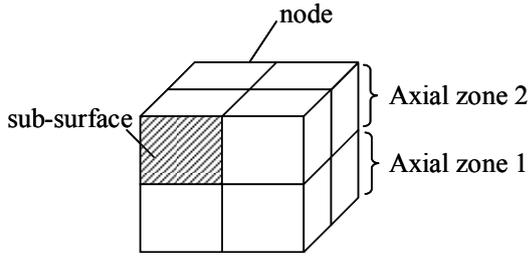


Fig. 1 An axially divided node

trons enter or leave is not only the mesh obtained by dividing the nodal surfaces but also the angle split. Although Eq. (1) includes a summation that continues for an infinite number of generations, the result will be saturated as the generation grow in number.²⁾ Thus, the summation of equations can be limited to a practical number of generations.

2. Matrix Formalization and Computing Acceleration Algorithm

(1) Matrix Formalization of 3D-DRM

Equation (1) which calculates each element is transformed to the matrix calculation equation. Equation (1) is rewritten as

$$R(k) = T + \frac{1}{k} SL + \frac{1}{k^2} SAL + \frac{1}{k^3} SA^2 L + \dots \quad (2)$$

Assuming that the inverse matrix of $kE - A$ exists, Eq. (2) is transformed using an inverse matrix as

$$R(k) = T + S(kE - A)^{-1} L, \quad (3)$$

where E means the identity matrix. However, reflector nodes have no fission reactions. Response matrices R_{ref} are equivalent to T and independent of k only in reflector nodes:

$$R_{ref} = T. \quad (4)$$

Incoming and outgoing neutron currents are related by using the response matrix as

$$J^{out} = R(k) J^{in}, \quad (5)$$

where J means neutron current and it is a one dimensional vector which consists of a series of neutron currents corresponding to sub-surfaces and energy groups. The k is evaluated under the condition where overall incoming and outgoing neutron currents are balanced. A pin-by-pin neutron production rate P can be evaluated directly as

$$P = S(kE - A)^{-1} J^{in}. \quad (6)$$

P is also a one dimensional vector.

(2) Matrix Symmetry

Nodal surfaces and fuel pins in a node are divided into some axial zones as shown in Fig. 1. Since the sub response matrices are evaluated in the infinite system, there is no distinction between top and bottom. This means that the response matrix is the same as the response matrix for the

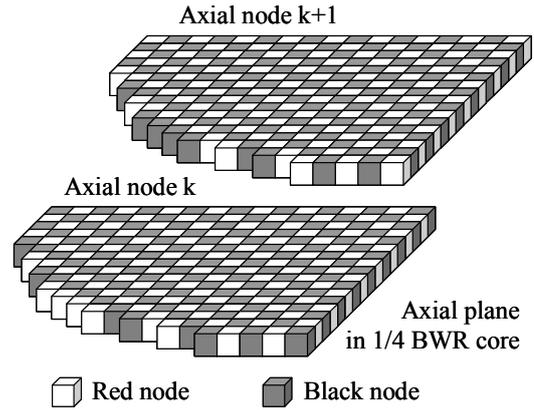


Fig. 2 The color sets for red-black method

node flipped upside down and it has symmetry. When a node is split vertically into upper and bottom sub-surfaces, the symmetry is described using block matrices as

$$\begin{pmatrix} R_1 & R_2 \\ R_2 & R_1 \end{pmatrix} = \begin{pmatrix} T_1 & T_2 \\ T_2 & T_1 \end{pmatrix} + \begin{pmatrix} S_1 & S_2 \\ S_2 & S_1 \end{pmatrix} \begin{pmatrix} B_1 & B_2 \\ B_2 & B_1 \end{pmatrix} \begin{pmatrix} L_1 & L_2 \\ L_2 & L_1 \end{pmatrix}, \quad (7)$$

$$\text{where } \begin{pmatrix} B_1 & B_2 \\ B_2 & B_1 \end{pmatrix} = \left(\begin{pmatrix} kE & 0 \\ 0 & kE \end{pmatrix} - \begin{pmatrix} A_1 & A_2 \\ A_2 & A_1 \end{pmatrix} \right)^{-1}.$$

Equation (7) is expanded as follows:

$$R_1 = T_1 + (S_1 B_1 + S_2 B_2) L_1 + (S_1 B_2 + S_2 B_1) L_2, \quad (8)$$

$$R_2 = T_2 + (S_1 B_1 + S_2 B_2) L_2 + (S_1 B_2 + S_2 B_1) L_1,$$

$$B_1 = \left(kE - A_1 - A_2 (kE - A_1)^{-1} A_2 \right)^{-1}, \quad (9)$$

$$B_2 = (kE - A_1)^{-1} A_2 \left(kE - A_1 - A_2 (kE - A_1)^{-1} A_2 \right)^{-1}.$$

When referenced to the bottom sub-surfaces, R_1 represents responses to the bottom sub-surfaces and R_2 represents responses to the upper sub-surfaces. As a result of using this symmetry, the number of calculations is reduced by half.

(3) Three-Dimensional Red-Black Method

The core k -effective is obtained by iterating current calculations. The 3D-DRM method uses the red-black response matrix acceleration method^{6,7)} to reduce the number of iterations. This acceleration method divides core nodes into two different color sets. In the 3D-DRM method, the core nodes are alternately divided in three-dimensions as shown in Fig. 2.

At first, the out-going currents in red nodes are calculated and in-coming currents in black nodes are updated. Next, the out-going currents in black nodes are calculated. This method accelerates the current convergence by updating half of the currents in the core during the whole current calculation.

3. Parallelization with Message Passing Interface (MPI) and OpenMP

(1) Parallelization

Parallelization of a calculation code is needed in order to get fast calculations using many computer resources. There

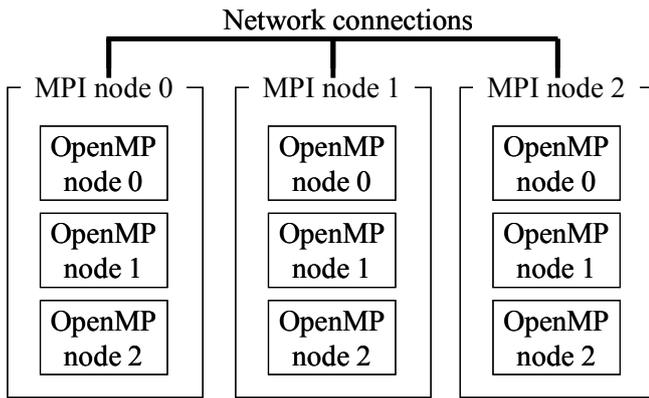


Fig. 3 The 3D-DRM parallelization scheme

are two different methods of programming for parallelization: MPI and OpenMP.^{4,5)} MPI is used for parallelization over network connections. OpenMP is used for parallelization within a computer node and sharing memory resources. Since the 3D-DRM method uses sub-response matrices which are made by lattice calculations, these matrices are used in common as nuclear data. OpenMP is suited to saving memory usage. However, MPI is needed for high performance computing because CPUs are limited in a computer node. The 3D-DRM method combines MPI and OpenMP. The parallelization scheme is shown in Fig. 3.

A MPI node corresponds to a physical computer node. An OpenMP node corresponds to a physical or virtual CPU. In the 3D-DRM method, the core is axially divided as indicated in Fig. 4 in order to use the same kinds of sub-response matrices in the same MPI node as much as possible because BWR fuel assemblies have axially different fuel enrichment.

(2) Automatic Load Balancing Method

In Eq. (3), the fuel nodes need to reconstruct the response matrices but the reflector nodes do not. Then, the MPI node which has more reflector nodes has less computational load. Since this unbalance causes computational inefficiency, the core nodes are divided automatically and evenly so that the computational load is uniform. Also, in PC clusters which have different performance, the load balancing method can maximize the efficiency.

4. 3D-DRM Core Simulator

The 3D-DRM core simulator consists of neutronic and thermal hydraulic calculation parts. The calculation flow chart is shown in Fig. 5. The parallelized section is only neutronic calculation part. The section has a nested iteration structure. The inner iteration for the current convergence is limited to 10 iterations. The outer iteration for the k-loop convergence is limited to 20 iterations. The automatic load balancing method measures the CPU time in neutronic calculation section and rearranges the core nodes after thermal hydraulic calculation. These calculations continue until convergence of the k-effective. In the end, the burn-up calculation performed. These neutronic and thermal hydraulic calculations go the next exposure step.

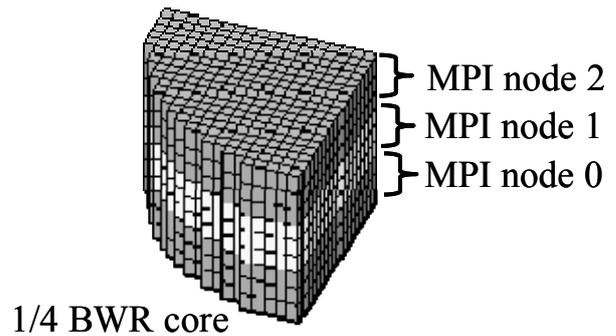


Fig. 4 Example of MPI node division

III. Numerical Results

A quarter core of the ABWR was analyzed with the coupled analysis of the neutronic and thermal hydraulic calculations by using the 3D-DRM method and its computing algorithms. The quarter core of the ABWR consists of 218 fuel assemblies. A 9×9 lattice fuel containing 66 full-length fuel rods, eight partial-length rods, and two large water rods was assumed. The calculation conditions for thermal-hydraulics such as the core thermal power, total coolant-flow, core pressure, and the thermal-hydraulic properties of the 9×9 lattice fuels were set to those in a practical plant. For calculation, the number of axial fuel nodes was 24 and the number of reflector nodes was 3 in each outer fuel node. The total number of nodes was 9,000.

In the 3D-DRM method, the sub-response matrices were calculated by the fuel assembly analysis code VMONT⁸⁾ which is based on a Monte Carlo neutron transport method. The VMONT code uses a multi-group model for the neutron spectrum calculation, and the total number of energy groups is 190. There were three energy groups of the produced sub-response matrices: Group 1 is from 5.53 keV to 10.0 MeV, Group 2 was from 0.625 eV to 5.53 keV, and Group 3 was from 0.0 to 0.625 eV. The number of tracked neutrons was 2×10^6 . The statistical uncertainty of the neutron infinite multiplication factor was about 0.03% Δk and that of the fuel rod neutron production rate was about 0.3%. The number of tracked neutrons was set so that the statistical uncertainty (one standard deviation) of the neutron infinite multiplication factor was within about 0.03% Δk . Each surface was subdivided by 4 for transverse segments, by 4 for angular segments and by 4 axial zones in a node.

The core calculation was performed in 12 PCs which had Core2 Quad Q9650 3 GHz (4 cores) and 8 GB memory. The network connection interface was gigabit Ethernet.

Figure 6 shows the k-effective difference for k iterations during one thermal-hydraulic loop in the 3D-DRM method with or without the red-black method. The method without the red-black method indicates the method which all in-coming and out-going currents are updated simultaneously. The k-effective difference means the difference from the k-effective value in convergence. Then, as the number of iterations increases, the k-effective difference decreases. The 3D-DRM method with the red-black method had 33% fewer

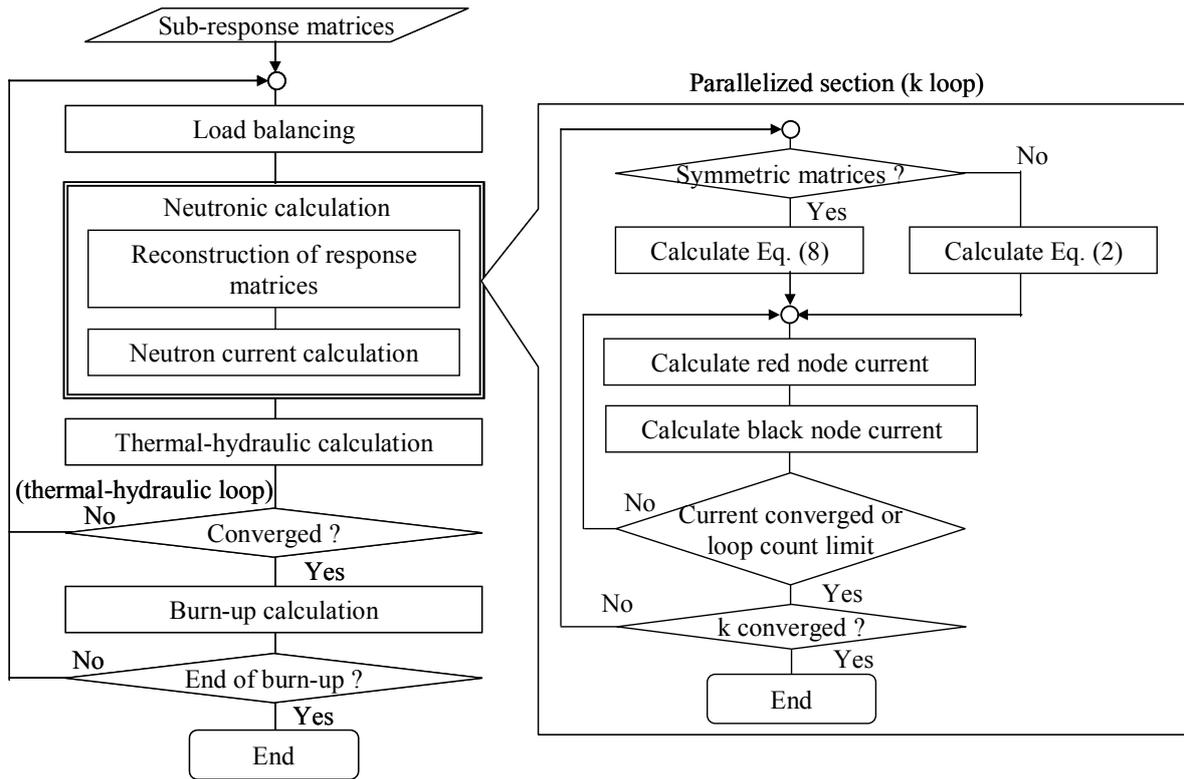


Fig. 5 Flow chart of parallelized 3D-DRM core simulator

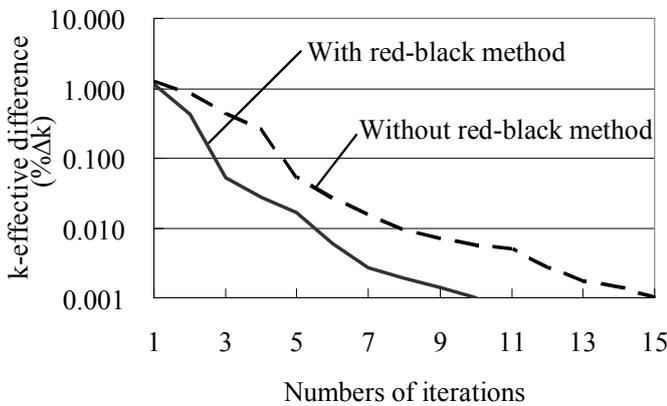


Fig. 6 Difference of k-effective for the number of iterations

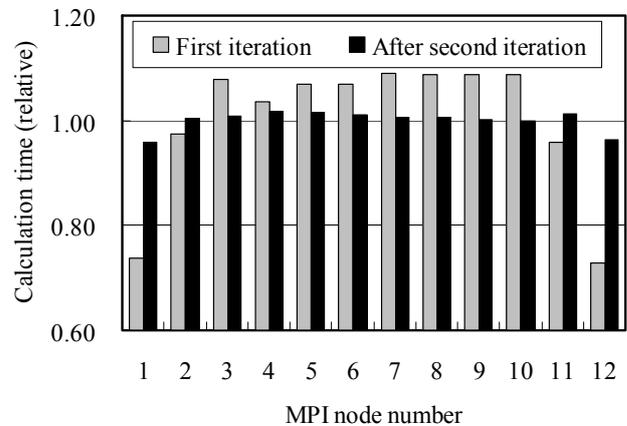


Fig. 7 The results for the load balancing method

numbers of iterations compared to the method without the red-black method.

The results of the automatic load balancing method are shown in **Fig. 7**. One iteration means one power and void coupled iteration. In the first iteration, the core nodes were evenly allocated to MPI nodes. The part of the core nodes in an axial layer is allocated to an MPI node for the uniform allocation. MPI node numbers 1 and 12 had shorter calculation times because these nodes which allocated upper or lower reflector nodes had more reflector nodes. In the second iteration, the automatic load balancer allocated more core nodes to MPI node numbers 1 and 12. As a result of load balancing, the calculation time was reduced to 10% of that after the first iteration.

The results of speedups in the PC cluster are shown in **Fig. 8**. The actual speedup was less than the ideal efficiency. That is why the parallelization code has some non-parallelized routines and a larger number of parallel computers cause shorter calculation times in parallel routines and relatively longer times in non-parallelized routines: this effect is called Amdahl's law.⁹⁾ Moreover, since more MPI divisions lead to more network communication time, the actual efficiency was lower with more nodes. In the 3D-DRM method, the speedup was about 8.5 in 12 PCs.

The total calculation time for an ABWR quarter core power and void iteration was about 4.5 hours with 12 PCs. The parallelization for the 3D-DRM method worked effectively.

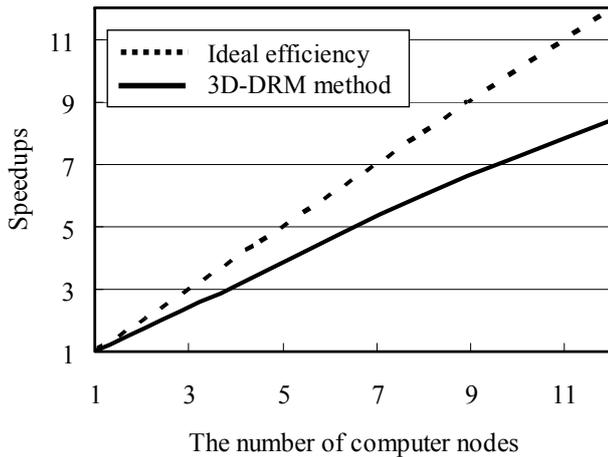


Fig. 8 Speedups for parallel computer nodes

IV. Conclusion

The 3D-DRM method and its computing algorithms were developed. The reconstruction of response matrices used sub-response matrices and the formalization with an inverse matrix. This formalization represented the neutron current and fuel pin power as one dimensional vectors. Moreover, using this matrix formalization, the matrix symmetry using block matrices was defined. This symmetry reduced the number of calculation by half. The 3D-DRM parallelization was performed using MPI and OpenMP to take advantage of the shared memory effectively and to have a larger number of parallel CPUs. In parallelization, two computing algorithms, the three-dimensional red-black method and the automatic load balancing method, were introduced. The red-black method reduced the number of k-effective itera-

tions by 33% and the automatic load balancing method also reduced the calculation time by 10%. As the result of implementing the parallelization and using the algorithms, the total calculation time for an ABWR quarter core power and void iteration was about 4.5 hours with 12 PCs.

References

- 1) M. Moriwaki, K. Ishii, H. Maruyama, M. Aoyama, "A new direct calculation method of response matrices using a Monte Carlo calculation," *J. Nucl. Sci. Technol.*, **36**[10], 877-887 (1999).
- 2) K. Ishii, T. Hino, T. Mitsuyasu, M. Aoyama, "Three-dimensional direct response matrix method using a Monte Carlo calculation," *J. Nucl. Sci. Technol.*, **46**[3], 259-267 (2009).
- 3) T. Hino, K. Ishii, T. Mitsuyasu, M. Aoyama, "Development of Pin-by-pin BWR Core Simulator using three-dimensional direct response matrix," *J. Nucl. Sci. Technol.*, **47**[5], 482-491 (2010).
- 4) The Message Passing Interface (MPI) standard, <http://www.mcs.anl.gov/research/projects/mpi/>
- 5) The OpenMP API specification for parallel programming, <http://openmp.org/wp/>
- 6) E. Lewis, G. Palmiotti, "Red-Black Response Matrix Acceleration by Transformation of Interface Variables," *Nucl. Sci. Eng.*, **130**, 181-193 (1998).
- 7) A. Yamamoto, "Acceleration of Response Matrix Method Using Cross-Section Scaling," *Nucl. Sci. Eng.*, **147**, 176-184 (2004).
- 8) Y. Morimoto, H. Maruyama, K. Ishii, M. Aoyama, "Neutronic analysis code for fuel assembly using a vectorized Monte Carlo method," *Nucl. Sci. Eng.*, **103**, 351-358 (1989).
- 9) G. Amdahl, "Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities," *AFIPS Conference Proceedings*, **30**, 483-485 (1967). <http://www-inst.eecs.berkeley.edu/~n252/paper/Amdahl.pdf>.