Bibliography

[Alexander, 1977] R. Alexander. Diagonally implicit Runge-Kutta methods for stiff ordinary differential equations, SIAM J. Numeric. Analysis, 30, 428-451, 19977.

[Borzi, 2002] A. Borzi. Fast multigrid methods for parabolic optimal control problems, Proc. Of the 18th GAMM-Seminar Leipzig, 1-10, 2002.

[Evtushenko, 1997] Yuri Evtushenko. Computation of exact gradients in distributed dynamic systems, Int. J. Control, 6(1), 81—89,1997.

[Farag, 2003] M. H. Farag. A gradient-type optimization technique for the optimal control for Schrödinger equations, I. Tech., Bulgaria, 16-25, 2003.

[Farag, 2004] M. H. Farag. A stability theorem for a class of distributed parameter control systems, accepted in Rocky Mountain J. of mathematics, 2004.

[Lion, 1972] J.-L Lions. Optimal control by systems described by partial differential equations, Mir, Moscow, SSSR, 1972.

[Farag, 2004] M. H. Farag. A stability theorem for a class of distributed parameter control systems, accepted in Rocky Mountain J. of mathematics, 2004.

[Shamardan] Ali B. Shamardan, M. H. Farag. and S. M. Refaat. Spatial central finite difference with DIRK(3,3) temporal schemes for nonlinear dispersive waves, Ricerche di Matematica, Vol. XLVII, fase 2,383-397,1998.

[Wu,2003] Y. H. Wu, M. Chuedoung and P. F. Siew. Optimal boundary control of a linear parabolic evaluation systems, Austra. Math. Soc., 123-134,2003.

Authors' Information

Mahmoud H. Farag – Department of Mathematics and Computer Science, Faculty of Education, Ibri, Sultanate of Oman, e-mail: farag5358@yahoo.com

Mainouna S.M. Al-Manthari – Department of Mathematics and Computer Science, Faculty of Education, Ibri, Sultanate of Oman, e-mail: may78may@hotmail.com

THE MATRIX METHOD OF DETERMINING THE FAULT TOLERANCE DEGREE OF A COMPUTER NETWORK TOPOLOGY

Sergey Krivoi, Miroslaw Hajder, Pawel Dymora, Miroslaw Mazurek

Abstract: This work presents a theoretical-graph method of determining the fault tolerance degree of the computer network interconnections and nodes. Experimental results received from simulations of this method over a distributed computing network environment are also presented.

Keywords: computer network, fault tolerance, coherent graph, regular graph, network topology, adjacency matrix.

ACM Classification Keywords: C.2.1 Network Architecture and Design - network topology, F.2.1 Numerical Algorithms and Problems - matrix methods, B.8.1 Reliability, Testing, and Fault-Tolerance - fault tolerance degree

Introduction

Computer networks plays an extremely important role in today's information technologies, because by its means it's possible to accelerate processes like i.e. transmission, processing and storage of information in computer systems. In such a process the most crucial issues are related with protecting a correct work of a computer network and its interconnection and node fault tolerance. The solution of these problems is related with examining the network topological characteristics and its topological structures. In this work the theoretical-graph method of determining the computer network topology critical points which refers to computer network interconnections and nodes failures is proposed.

1. Preliminary Information and Definitions

Computer network (CN) consisted of $n > 1$ computers connected between themselves is presented as a graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ represents a real number of nodes and $E = \{(u, v): u, v \in V\}$ – a real number of interconnections. Under these assumptions, the network nodes are represented by graph G nodes, and network interconnections corresponds with the given graph connection links. If not assumed differently a graph definition is always meant as an undirected graph. If $e = (u, v) \in E$, than node u (node v) is called an end of a link e, and such nodes – adjacent. If node u turns out to be the end of link e, than link e and node u, is called incident. Note, that the adjacency relation for a graph node turns out to be symmetric. If $u \in V$, than $n(u)$ a number of graph links incidental to node u is called a degree of a node u. A path from a node u to a node v of the graph $G = (V, E)$ with a length k is called a link sequence $(u_1, u_2), \ldots, (u_k, u_{k+1}) \in E$ such as $u = u_1$, $v = u_{k+1}$.

Definition 1. Graph $G = (V, E)$ is called coherent if from any node u there exists a path to its any other node v (symmetric-relation – inverted path: from a node v to a node u).

An operation of link and node removal is considered on examples of a Cartesian product and an isomorphic joint of two graphs.

Graph $G' = (V', E') = G - v$ is called a graph received from a graph $G = (V, E)$, as a result of applying a node $v \in V$ removal operation if $V' = V \setminus \{v\}$, and E' squares with a number E, from which all links incident to a node v were removed.

Graph $G' = (V', E') = G - e$ is named a graph received from a graph $G = (V, E)$ as a result of applying a link $e \in E$ removal operation, if $V' = V$, and $E' = E \setminus \{e\}$.

Notice, that for both operations the following equations are true:

$$(G - u) - v = (G - v) - u; \quad (G - e) - e' = (G - e') - e,$$

i.e. the result graphs doesn't depend on the sequence of links or nodes removal order .

The nodes number M (or links) of a graph $G = (V, E)$ is called crucial, if as a result of its elements removal from a given graph this graph becomes incoherent.

Let graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be given. Graph $G = (V, E) = G_1 \times G_2$ is called a graph received as a result of applying a Cartesian product operation on graphs G_1 and G_2, if $V = V_1 \times V_2$ and $E = \{[(u, u'), (v, v')]: u = v$ in a graph G_1 and $(u', v') \in E_2$ or $u' = v'$ in a graph G_2 and $(u, v) \in E_1\}$. As an example, if this operation is applied on graphs presented in the following figure 1 the result graph is called a toroid.

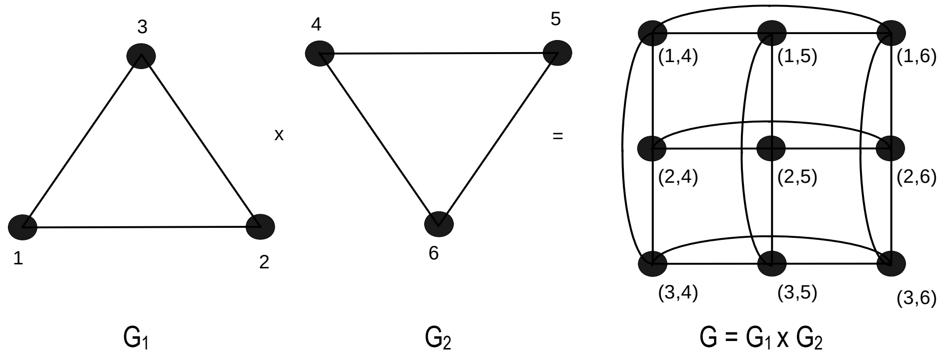

Fig. 1. Example of a Cartesian product $G_1 \times G_2$ operation

Definition 2. Graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are called isomorphic, if between the sets of nodes V_1 and V_2 exists a bijection mapping $f: V_1 \rightarrow V_2$, such that nodes u and v are adjacent to the graph G_1, than nodes $f(v)$ and $f(v)$ are adjacent in the graph G_2. Mapping f is called isomorphic.

Let isomorphic graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ and f – isomorphic mapping are given.

Graph $G = (V, E) = G_1 *_f G_2$ is called a graph received as a result of applying an isomorphic joint operation on graphs G_1 and G_2, if $V = V_1 \cup V_2$ and also $E = E_1 \cup E_2 \cup \{(u, u'): u \in V_1, u' \in V_2$ and $f(u) = u'\}$.

As an example, if this operation is applied on graphs presented in the figure 2 with isomorphic $f(i) = i + 8$ the result graph is called a hypercube.

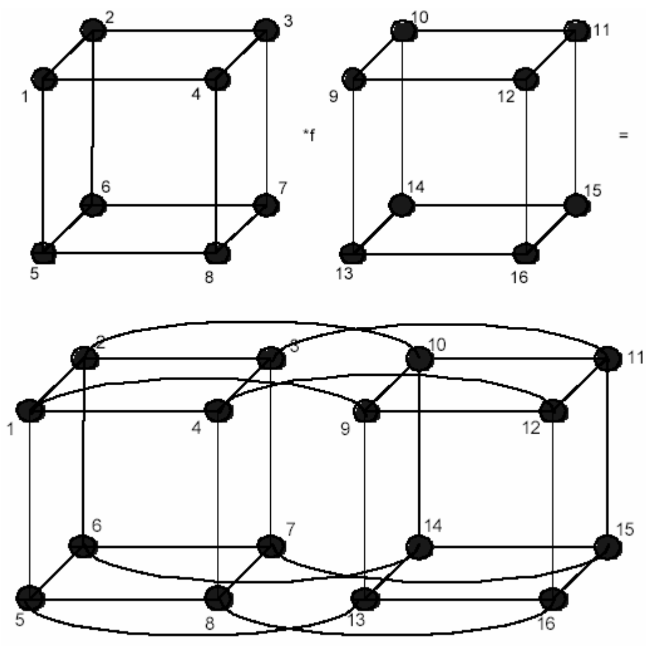

Fig. 2. Example of an isomorphic $G_1 f G_2$ operation

The next theorem is following immediately from definitions.

Theorem 1. If graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ – are coherent, than a graph received as a result of applying a Cartesian product operation or an isomorphic joint operation is also coherent.

It's obvious that mentioned links and nodes removal operations in coherent graphs may create incoherent graphs. With each graph $G = (V, E)$ a matrix $A_G = ||a_{ij}||$ is related, where $i, j = 1, 2, ..., n$, and is called an adjacency matrix and is described as follows:

$$a_{ij} = \begin{cases} 1, \text{ if } (v_i, v_j) \in E, \\ 0, \text{ otherwise.} \end{cases}$$

Therefore, graph G with n nodes corresponds to a square matrix $n \times n$, filled up with 0 and 1 values. Introduced links and nodes removal operations in graphs may be carried out in an easy manner on these graphs adjacency matrixes.

2. The Matrix Method of Determining the Fault Tolerance Degree of a CN

Let a CN consists of n computers and is represented as a graph $G = (V, E)$. With a help of a graph G analysis for a given CN the most critical points are specified. That's why formal statements are introduced.

Let $G = (V, E)$ – is the given coherent graph representing CN, while A_G – is the adjacency matrix of this graph. The nodes number $V' \subseteq V$ (links $E' \subseteq E$) of a graph $G = (V, E)$, is called a computer critical point (failure place) if V' (E') the minimal nodes set (links) of a graph G becomes crucial, than any element removal from the graph G causes that this graph becomes incoherent.

Fulfilling these definitions and an adjacency matrix of a graph G, representing a CN, it's possible to create a method of determining the critical points in a CN. In this end the matrix interpretation of introduced before graph operation is considered.

2.1. The Matrix Interpretation of the Graph Operations

As follows from determining critical points this method turns out to be useful in finding the minimal existing subsets. In order to do this a graph adjacency matrix is used because for an inherent graph its adjacency matrix becomes a diagonally-block. With a reorder of an appropriate rows and columns it becomes as follows:

$$A_G = \begin{bmatrix} A \mid 0 \\ 0 \mid B \end{bmatrix},$$

where *A* and *B* are nonzero matrixes, and 0 – represents zero matrixes. On the basis of this simple adjacency matrix property, the matrix method of determining the critical points of a CN is based.

Interpretation of nodes and links removal operation in an adjacency matrix turns out to be very simple. In fact, the removal of links (v_i, v_j) in a graph *G* reduces to changing of an element a_{ij} values from 1 to 0 in a matrix A_G, and a removal of a node v_i in a graph *G* corresponds with a removal of an *i*-ary row of this matrix.

For example, for a graph presented in the figure 3a its adjacency matrix has such a structure (rows and columns of the matrix corresponds to enumeration 1,2,3,4,5,6 from the left to the right and from up to down):

$$A_G = \begin{vmatrix} 011000 \\ 101110 \\ 110000 \\ 010011 \\ 010101 \\ 000110 \end{vmatrix}.$$

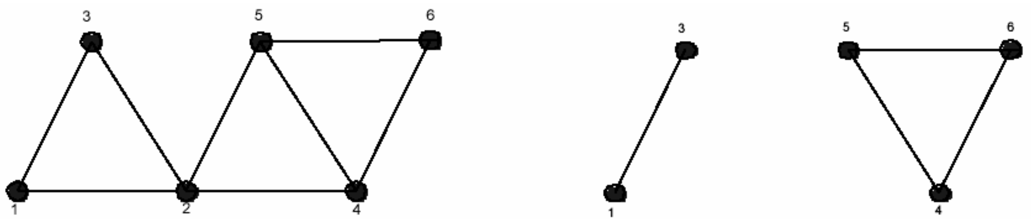

Fig. 3. a) Example of the graph with 6 nodes; b) Graph with removed links and nodes.

The node 2 removal from this graph reduces to a matrix $A_G = \begin{vmatrix} 01000 \\ 10000 \\ 00011 \\ 00101 \\ 00110 \end{vmatrix}$, for which graph is shown in the Fig. 3b.

The Cartesian product operation on graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ on the basis of an adjacency matrix is possible to present on an example. From this operation definition it results that an adjacency matrix of a graph $G = G_1 \times G_2$ has a following form:

$$A_G = \begin{vmatrix} A_{G_1} \mid E_a \mid E_a \\ E_a \mid A_{G_1} \mid E_a \\ E_a \mid E_a \mid A_{G_1} \end{vmatrix},$$

where a size of an adjacency matrix A_{G1} is equal to $|V_2|$, while E_a - a diagonal matrix of the size *a* which on its diagonal accepts values 0 or 1 depending on whether the other pair nodes in a graph G_2 are adjacent to it or not.

For example, an adjacency matrix of a graph $G = G_1 x G_2$, where graphs G_1 and G_2 are presented in the figure 1, has a following form (matrix rows and columns correspond to an enumeration (1,4), (2,4), (3,4), (1,5), (2,5), (3,5), (1,6), (2,6), (3,6)):

$$A_G = \begin{vmatrix} 011 \mid 100 \mid 100 \\ 101 \mid 010 \mid 010 \\ 110 \mid 001 \mid 001 \\ 100 \mid 011 \mid 100 \\ 010 \mid 101 \mid 010 \\ 001 \mid 110 \mid 001 \\ 100 \mid 100 \mid 011 \\ 010 \mid 010 \mid 101 \\ 001 \mid 001 \mid 110 \end{vmatrix}.$$

The isomorphic joint $G = G_1 *_f G_2$, of graphs G_1 and G_2 has an adjacency matrix as it follows from the applied operation:

$$A_G = \begin{vmatrix} A_{G_1} \mid C \\ C \mid A_{G_2} \end{vmatrix},$$

where in a diagonal the adjacency matrixes of graphs G_1 and G_2 are placed respectively, and an element a_{ij} of a matrix C is equal 1, if $f(v_i) = v_j$, or equal 0 otherwise. For example, if this operation is applied on a graph from the figure 2, then an adjacency matrix of this graph has a following form (see on the right side of the text):

$$A_G = \begin{vmatrix} 01011000 \mid 10000000 \\ 10100100 \mid 01000000 \\ 01010010 \mid 00100000 \\ 10100001 \mid 00010000 \\ 10000101 \mid 00001000 \\ 01001010 \mid 00000100 \\ 00100101 \mid 00000010 \\ 00011010 \mid 00000001 \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ 10000000 \mid 01011000 \\ 01000000 \mid 10100100 \\ 00100000 \mid 01010010 \\ 00010000 \mid 10100001 \\ 00001000 \mid 10000101 \\ 00000100 \mid 01001010 \\ 00000010 \mid 00100101 \\ 00000001 \mid 00011010 \end{vmatrix}$$

2.2. The Matrix Method of the CN Topology Analysis

Let $G = (V, E)$ – represents a coherent graph describing a CN and A_G – the adjacency matrix of this graph. As follows, with a simple analysis of a graph adjacency matrix it's possible to determine critical points of a CN, which is represented by this graph.

Generally, in a simple way it's possible to determine nodes, which has the lowest degree because the numbers of 1 in a row (or column) is equal to a node degree, which corresponds with this row (or column). After the nodes with the minimal degree are chosen then the starting number of links which might be suspected of critical links point are received. However, it's not guaranteed that they will appear because the minimal degree doesn't causes that the number of incident to them links is the minimal existing number. Suspected of number of nodes (links) is removed from the adjacency matrixes and after this, a received matrix has a block form. If a matrix has such a form then removed number of nodes (links) turns out to be crucial. Such an analysis requires a consideration of 2^n variants.

The general algorithm, which results from the above-presented analysis, might be presented in a following way (A – an adjacency matrix of a graph *G*, and *n* – a sequence presenting a CN):

Critical points (A)

1. Determine critical points *M* of a graph *G*, consisted of one element of a matrix *A*.
2. for $i = 2$ to 2^n fulfill
   Determine critical points *M'* of a graph *G*, consisted of *i* elements of a matrix *A*
   If $/M'/ < /M/$ than $M:=M'$;
   End of cycle.
3. Return *(M)*.

The first operator is processed directly from the same matrix *A* (the before presented manner), while the second operator << determines critical points *M'* of a graph *G*, consisted of *i* elements of a matrix *A,* >> presents by itself computations of a transition closure of an achievement relation in a graph. It's common knowledge that a given algorithm is based on an estimation of *n-i-x* (or n) matrix degree received from the matrix *A* as a result of its node (or link) removal.

Obviously, a cycle described in a second paragraph can be parallelized because computations of different iterations are independent from each other. In the following paragraph an experimental results of computations on a cluster with 10 processors are presented.

3. Experimental Results

In order to parallel ours computations the Parallel Virtual Machine (PVM) was used. Next, the PVM was implemented on a cluster with 10 processors. The PVM is a main parallel library, which may process on heterogeneous computer networks. Created in PVM the Message Passing Interface (MPI) which is a library of procedures and functions became a modern standard of building parallel applications. The MPI is independent of operating systems' platform.

Nodes number	Time [s]
5	0,0028
10	0,0047
50	0,0228
100	0,0431
250	0,34
500	0,6402
750	0,7913
1000	1,4252
1500	2,1958
2500	3,47
5000	7,3218
7500	8,9551
10000	22,8534

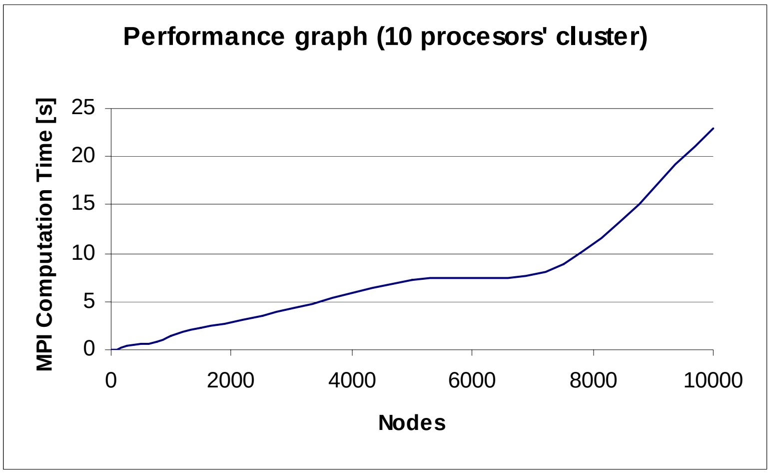

Fig. 4. Experimental results

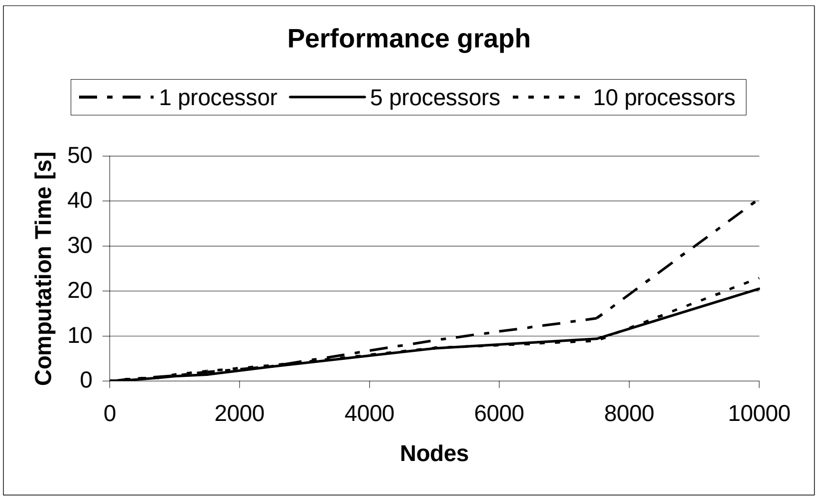

Fig. 5. Experimental results

Conclusions

The presented method of determining the critical points is based on the graph theory and some features of the adjacency matrix, which represents graphs. Searching for critical points in computer networks, as it follows from the above researches is characterized with a large complexity and requires applying of a great computational performance. However, the main advantage of this method is a fact that is uses homogenous structures, and the computations itself are of the same type. Presented experiments were implemented and realized on the multiprocessor cluster and the results of these experiments are presented in the above table. Analyzing these experimental results, the conclusion may be drawn independently.

Bibliography

[Капитонова Ю.В., Кривой С.Л., 1] Капитонова Ю.В., Кривой С.Л., Летичевский А.А., Луцкий Г.М. – *Лекции по дискретной математике*. – Санкт-Петербург: БХВ-Петербург. –2004. 614 стр.

[Свами М., Тхуласираман К., 2] Свами М., Тхуласираман К. Графы, сети и алгоритмы. Москва: Мир. – 1984. 454 стр.

[Hajder M., Loutski G., 3] Hajder M., Loutskij G., Streciwilk W. - *Informatyka*. Rzeszow: WsiZ. - 2002. – 714 s. (на польском языке)

[Hajder M., Dymora P., 4] Hajder M., Dymora P. - *Algorithmical and topological methods of fault tolerace assurance*; Wydawnictwo Uniwersytetu Marii Curie-Skłodowskiej; Annales Informatica, Kazimierz Dolny 2004 – 143-153,

[Hajder M., Dymora P., Bolanowski M., 5] M. Hajder, P. Dymora, M. Bolanowski: *Fault tolerance in multichannel scalability systems;* International Conference of Science and Technology Scalnet'04, Scalable Systems and Computer Networks Design and Aplications, Ministry of Education and Science of Ukraine Kremenchuk State Polytechnical University, Kremenchuk 2004, – pp. 52-57,

[Spohn D.,L., 6] Spohn D. L. - *Data Network Design. New York: McGrew-Hill*. – 1997. – 347 p.

[Harsall F., 7] Harsall F. - *Data Communications, Computer Design Networks and Open Systems*. 4-th Eddition. – Addison-Wesley Company. – Harlow. – 1996. – 412 p.

[Mazurek M., 8] Mazurek M. - *Hyper-deBruijn topology as a result of the Network Product*, International Conference of Science and Technology Scalnet'04, Scalable Systems and Computer Networks Design and Aplications, Ministry of Education and Science of Ukraine Kremenchuk State Polytechnical University, Kremenchuk 2004, pp. 62-65.

[Hajder M., Mazurek M., 9] Hajder M., Mazurek M. - *Embending as a method to improve multichanel bus topology parameters*, Annales Universitatis Mariae Curie-Skłodowska, Sectio Ai Informatica, Lublin, 2004.

Authors' Information

Krivoi Sergey – Glushkov Institute of Cybernetics NAS of Ukraine, Ukraine, Kiev, 03187, 40 Glushkova Street, Institute of Cybernetics, e-mail: krivoi@i.com.ua

Hajder Miroslaw – Technical University of Rzeszow, Rzeszow, Poland, e-mail: mhajder@prz.rzeszow.pl

Dymora Pawel – Technical University of Rzeszow, Rzeszow, Poland, e-mail: pdymora@prz.rzeszow.pl

Mazurek Miroslaw – Technical University of Rzeszow, Rzeszow, Poland, e-mail: mmazurek@prz.rzeszow.pl

RKHS-METHODS AT SERIES SUMMATION FOR SOFTWARE IMPLEMENTATION

Svetlana Chumachenko, Ludmila Kirichenko

Abstract: Reproducing Kernel Hilbert Space (RKHS) and Reproducing Transformation Methods for Series Summation that allow analytically obtaining alternative representations for series in the finite form are developed.

Keywords: The reproducing transformation method, Hilbert space, reproducing kernel, RKHS, Series Summation Method.

ACM Classification Keywords: G.1.10 Mathematics of Computing: Applications