



Introduction to numerical analyses

Laith adnan hadi almhana
almhanalaith08@gmail.com

Abstract

Delay differential equations (DDEs) are a class of differential equations that have received considerable recent attention and been shown to model many real life problems, traditionally formulated as systems of ordinary differential equations (ODEs), more naturally and more accurately. By the development and progress of science in the recent period another images emerged from these equations, such as: neutral delay differential equations (NDDEs), Volterra delay-integro-differential equations (VDIDEs), and neutral Volterra delay-integro-differential equations (NVDIDEs). As a result of the difficulty of finding analytical solutions to these equations we often resort to numerical methods to solve these problems. The spline collocation methods, that depend on the second and higher derivatives, for solving DDEs have been considered by several authors. These methods produce less accurate approximations when the solution of these equations has a discontinuity in the second and higher derivatives. The aim of this thesis is to present a class of spline collocation methods which are based on the first derivative for the numerical solution of DDEs, NDDEs and VDIDEs. When the solution of these equations has a discontinuity in the second and higher derivatives, the present methods produce numerical solutions better than those computed by the previous methods. We present a combination of these methods.

Keywords: numerical analyses

مقدمة في التحليلات العددية

ليث عدنان هادي المحنة
almhanalaith08@gmail.com

خلاصة

المعادلات التفاضلية المؤجلة (DDEs) هي فئة من المعادلات التفاضلية التي حظيت باهتمام كبير مؤخرًا وتبين أنها تمثل العديد من مشاكل الحياة الحقيقية، والتي تمت صياغتها تقليديًا كأنظمة للمعادلات التفاضلية العادية (ODEs)، بشكل أكثر طبيعية وأكثر دقة. ومع تطور العلم وتقدمه في الفترة الأخيرة ظهرت صور أخرى لهذه المعادلات، مثل: معادلات تفاضلية تأخير تكاملية محايدة (NDDEs)، ومعادلات فولتيرا تفاضلية تأخيرية تكاملية (VDIDEs)، ومعادلات فولتيرا تفاضلية تأخيرية (NVDIDEs). ونتيجة لصعوبة إيجاد الحلول التحليلية لهذه المعادلات فإننا كثيراً ما نلجأ إلى الطرق العددية لحل هذه المسائل. لقد تم دراسة طرق جميع الشرائح، التي تعتمد على المشتقات الثانية والأعلى، لحل DDEs من قبل العديد من المؤلفين. تنتج هذه الطرق تقديرات تقريبية أقل دقة عندما يكون حل هذه المعادلات به انقطاع في المشتقات الثانية والأعلى. الهدف من هذه الأطروحة هو تقديم فئة من طرق جميع الشرائح والتي تعتمد على المشتق الأول



للحل العددي لـ DDEs و NDDEs و VDIDES. عندما يكون حل هذه المعادلات به انقطاع في المشتقات الثانية والمشتقات العليا فإن الطرق الحالية تنتج حلولاً عددية أفضل من تلك المحسوبة بالطرق السابقة. نقدم مجموعة من هذه الأساليب.

كلمات مفتاحية : التحليلات العددية

1-Introduction

Numerical analysis, area of mathematics and computer science that creates, analyzes, and implements algorithms for obtaining numerical solutions to problems involving continuous variables. Such problems arise throughout the natural sciences, social sciences, engineering, medicine, and business. Since the mid-20th century, the growth in power and availability of digital computers has led to an increasing use of realistic mathematical models in science and engineering, and numerical analysis of increasing sophistication is needed to solve these more detailed models of the world. The formal academic area of numerical analysis ranges from quite theoretical mathematical studies to computer science issues[1].

With the increasing availability of computers, the new discipline of scientific computing, or computational science, emerged during the 1980s and 1990s. The discipline combines numerical analysis, symbolic mathematical computations, computer graphics, and other areas of computer science to make it easier to set up, solve, and interpret complicated mathematical models of the real world[2].

2-Common perspectives in numerical analysis

Numerical analysis is concerned with all aspects of the numerical solution of a problem, from the theoretical development and understanding of numerical methods to their practical implementation as reliable and efficient computer programs. Most numerical analysts specialize in small subfields, but they share some common concerns, perspectives, and mathematical methods of analysis. These include the following[3]:

1. When presented with a problem that cannot be solved directly, they try to replace it with a “nearby problem” that can be solved more easily. Examples are the use of interpolation in developing numerical integration methods and root-finding methods.
2. There is widespread use of the language and results of linear algebra, real analysis, and functional analysis (with its simplifying notation of norms, vector spaces, and operators).



3. There is a fundamental concern with error, its size, and its analytic form. When approximating a problem, it is prudent to understand the nature of the error in the computed solution. Moreover, understanding the form of the error allows creation of extrapolation processes to improve the convergence behaviour of the numerical method.

4. Numerical analysts are concerned with stability, a concept referring to the sensitivity of the solution of a problem to small changes in the data or the parameters of the problem. Consider the following example. The polynomial $p(x) = (x - 1)(x - 2)(x - 3)(x - 4)(x - 5)(x - 6)(x - 7)$, or expanded, $p(x) = x^7 - 28x^6 + 322x^5 - 1,960x^4 - 6,769x^3 - 13,132x^2 + 13,068x - 5,040$ has roots that are very sensitive to small changes in the coefficients. If the coefficient of x^6 is changed to -28.002 , then the original roots 5 and 6 are perturbed to the complex numbers $5.459 \pm 0.540i$ —a very significant change in values. Such a polynomial $p(x)$ is called unstable or ill-conditioned with respect to the root-finding problem. Numerical methods for solving problems should be no more sensitive to changes in the data than the original problem to be solved. Moreover, the formulation of the original problem should be stable or well-conditioned[4].

5. Numerical analysts are very interested in the effects of using finite precision computer arithmetic. This is especially important in numerical linear algebra, as large problems contain many rounding errors.

6. Numerical analysts are generally interested in measuring the efficiency (or “cost”) of an algorithm. For example, the use of Gaussian elimination to solve a linear system $Ax = b$ containing n equations will require approximately $2n^3/3$ arithmetic operations. Numerical analysts would want to know how this method compares with other methods for solving the problem[5].

3-Modern applications and computer software

Numerical analysis and mathematical modeling are essential in many areas of modern life. Sophisticated numerical analysis software is commonly embedded in popular software packages (e.g., spreadsheet programs) and allows fairly detailed models to be evaluated, even when the user is unaware of the underlying mathematics. Attaining this level of user transparency requires reliable, efficient, and accurate numerical analysis software, and it requires problem-solving environments (PSE) in which it is relatively easy to model a given situation. PSEs are usually based on excellent theoretical mathematical models, made available to the user through a convenient graphical user interface[6].

4-Applications



Computer-aided engineering (CAE) is an important subject within engineering, and some quite sophisticated PSEs have been developed for this field. A wide variety of numerical analysis techniques is involved in solving such mathematical models. The models follow the basic Newtonian laws of mechanics, but there is a variety of possible specific models, and research continues on their design. One important CAE topic is that of modeling the dynamics of moving mechanical systems, a technique that involves both ordinary differential equations and algebraic equations (generally nonlinear). The numerical analysis of these mixed systems, called differential-algebraic systems, is quite difficult but necessary in order to model moving mechanical systems. Building simulators for cars, planes, and other vehicles requires solving differential-algebraic systems in real time[7].

Another important application is atmospheric modeling. In addition to improving weather forecasts, such models are crucial for understanding the possible effects of human activities on the Earth's climate. In order to create a useful model, many variables must be introduced. Fundamental among these are the velocity $V(x, y, z, t)$, pressure $P(x, y, z, t)$, and temperature $T(x, y, z, t)$, all given at position (x, y, z) and time t . In addition, various chemicals exist in the atmosphere, including ozone, certain chemical pollutants, carbon dioxide, and other gases and particulates, and their interactions have to be considered. The underlying equations for studying $V(x, y, z, t)$, $P(x, y, z, t)$, and $T(x, y, z, t)$ are partial differential equations; and the interactions of the various chemicals are described using some quite difficult ordinary differential equations. Many types of numerical analysis procedures are used in atmospheric modeling, including computational fluid mechanics and the numerical solution of differential equations. Researchers strive to include ever finer detail in atmospheric models, primarily by incorporating data over smaller and smaller local regions in the atmosphere and implementing their models on highly parallel supercomputers. Get a Britannica Premium subscription and gain access to exclusive content[8].

-Subscribe Now

Modern businesses rely on optimization methods to decide how to allocate resources most efficiently. For example, optimization methods are used for inventory control, scheduling, determining the best location for manufacturing and storage facilities, and investment strategies[9].

-Computer software

Software to implement common numerical analysis procedures must be reliable, accurate, and efficient. Moreover, it must be written so as to be easily portable between different computer systems. Since about 1970, a number of government-



sponsored research efforts have produced specialized, high-quality numerical analysis software. The most popular programming language for implementing numerical analysis methods is Fortran, a language developed in the 1950s that continues to be updated to meet changing needs. Other languages, such as C, C++, and Java, are also used for numerical analysis. Another approach for basic problems involves creating higher level PSEs, which often contain quite sophisticated numerical analysis, programming, and graphical tools. Best known of these PSEs is MATLAB, a commercial package that is arguably the most popular way to do numerical computing. Two popular computer programs for handling algebraic-analytic mathematics (manipulating and displaying formulas) are Maple and Mathematica[10].

5-Historical background

Numerical algorithms are at least as old as the Egyptian Rhind papyrus (c. 1650 BC), which describes a root-finding method for solving a simple equation. Ancient Greek mathematicians made many further advancements in numerical methods. In particular, Eudoxus of Cnidus (c. 400–350 BC) created and Archimedes (c. 285–212/211 BC) perfected the method of exhaustion for calculating lengths, areas, and volumes of geometric figures. When used as a method to find approximations, it is in much the spirit of modern numerical integration; and it was an important precursor to the development of calculus by Isaac Newton (1642–1727) and Gottfried Leibniz (1646–1716)[11].

Calculus, in particular, led to accurate mathematical models for physical reality, first in the physical sciences and eventually in the other sciences, engineering, medicine, and business. These mathematical models are usually too complicated to be solved explicitly, and the effort to obtain approximate, but highly useful, solutions gave a major impetus to numerical analysis. Another important aspect of the development of numerical methods was the creation of logarithms about 1614 by the Scottish mathematician John Napier and others. Logarithms replaced tedious multiplication and division (often involving many digits of accuracy) with simple addition and subtraction after converting the original values to their corresponding logarithms through special tables. (Mechanization of this process spurred the English inventor Charles Babbage (1791–1871) to build the first computer—see History of computers: The first computer.)

Newton created a number of numerical methods for solving a variety of problems, and his name is still attached to many generalizations of his original ideas. Of particular note is his work on finding roots (solutions) for general functions and finding a polynomial equation that best fits a set of data (“polynomial



interpolation”). Following Newton, many of the mathematical giants of the 18th and 19th centuries made major contributions to numerical analysis. Foremost among these were the Swiss Leonhard Euler (1707–1783), the French Joseph-Louis Lagrange (1736–1813), and the German Carl Friedrich Gauss (1777–1855).

One of the most important and influential of the early mathematical models in science was that given by Newton to describe the effect of gravity. According to this model, the gravitational force exerted on a body of mass m by the Earth has magnitude $F = Gmm_e/r^2$, where m_e is the mass of the Earth, r is the distance between the centres of the two bodies, and G is the universal gravitational constant. The force on m is directed toward the centre of gravity of the Earth. Newton’s model has led to many problems that require solution by approximate means, usually involving ordinary differential equations[12].

Following the development by Newton of his basic laws of physics, many mathematicians and physicists applied these laws to obtain mathematical models for solid and fluid mechanics. Civil and mechanical engineers still base their models on this work, and numerical analysis is one of their basic tools. In the 19th century, phenomena involving heat, electricity, and magnetism were successfully modeled; and in the 20th century, relativistic mechanics, quantum mechanics, and other theoretical constructs were created to extend and improve the applicability of earlier ideas. One of the most widespread numerical analysis techniques for working with such models involves approximating a complex, continuous surface, structure, or process by a finite number of simple elements. Known as the finite element method (FEM), this technique was developed by the American engineer Harold Martin and others to help the Boeing Company analyze stress forces on new jet wing designs in the 1950s. FEM is widely used in stress analysis, heat transfer, fluid flow, and torsion analysis[13].

6-Theory of numerical analysis

The following is a rough categorization of the mathematical theory underlying numerical analysis, keeping in mind that there is often a great deal of overlap between the listed areas.

-Numerical linear and nonlinear algebra

Many problems in applied mathematics involve solving systems of linear equations, with the linear system occurring naturally in some cases and as a part of the solution process in other cases. Linear systems are usually written using matrix-vector notation, $Ax = b$ with A the matrix of coefficients for the system, x the column vector of the unknown variables $x_1, \dots, x_n$, and b a given



column vector. Solving linear systems with up to 1,000 variables is now considered relatively straightforward in most cases. For small to moderately sized linear systems (say, $n \leq 1,000$), the favoured numerical method is Gaussian elimination and its variants; this is simply a precisely stated algorithmic variant of the method of elimination of variables that is introduced in elementary algebra. For larger linear systems, there is a variety of approaches depending on the structure of the coefficient matrix A . Direct methods lead to a theoretically exact solution x in a finite number of steps, with Gaussian elimination the best-known example. In practice, there are errors in the computed value of x due to rounding errors in the computation, arising from the finite length of numbers in standard computer arithmetic. Iterative methods are approximate methods that create a sequence of approximating solutions of increasing accuracy[14].

Nonlinear problems are often treated numerically by reducing them to a sequence of linear problems. As a simple but important example, consider the problem of solving a nonlinear equation $f(x) = 0$. Approximate the graph of $y = f(x)$ by the tangent line at a point $x^{(0)}$ near the desired root (use of parentheses is a common notational convention to distinguish successive iterations from exponentiation), and use the root of the tangent line to approximate the root of the original nonlinear function $f(x)$. This leads to Newton's iterative method for finding successively better approximations to the desired root: $x^{(k+1)} = x^{(k)} - f(x^{(k)})/f'(x^{(k)})$, $k = 0, 1, 2, \dots$, where $f'(x)$ indicates the first derivative of the original function.

This generalizes to handling systems of nonlinear equations. Let $f(x) = 0$ denote a system of n nonlinear equations in n unknowns $x = (x_1, \dots, x_n)$. Newton's method for solving this system is given by $x^{(k+1)} = x^{(k)} + \delta^{(k)}$, where $f'(x^{(k)})\delta^{(k)} = -f(x^{(k)})$, $k = 0, 1, 2, \dots$

In this, $f'(x)$ is a generalization of the derivative known as the Jacobian matrix of $f(x)$, and the second equation is a linear system of order n . There are numerous other approaches to solving nonlinear systems, most based on using some type of approximation involving linear functions.

An important related class of problems occurs under the heading of optimization. Given a real-valued function $f(x)$ with x a vector of unknowns, a value of x that minimizes $f(x)$ is sought. In some cases x is allowed to vary freely, and in other cases there are constraints on x . Such problems occur frequently in business applications.

7-Approximation theory



This category includes the approximation of functions with simpler or more tractable functions and methods based on using such approximations. When evaluating a function $f(x)$ with x a real or complex number, it must be kept in mind that a computer or calculator can only do a finite number of operations. Moreover, these operations are the basic arithmetic operations of addition, subtraction, multiplication, and division, together with comparison operations such as determining whether $x > y$ is true or false. With the four basic arithmetic operations, it is possible to evaluate polynomials $p(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ as well as rational functions (polynomials divided by polynomials). By including the comparison operations, it is possible to evaluate different polynomials or rational functions on different sets of real numbers x . The evaluation of all other functions—e.g., $f(x) = \text{Square root of } x$ or 2^x —must be reduced to the evaluation of a polynomial or rational function that approximates the given function with sufficient accuracy. All function evaluations on calculators and computers are accomplished in this manner[15].

One common method of approximation is known as interpolation. Consider a set of points (x_i, y_i) where $i = 0, 1, \dots, n$, and then find a polynomial that satisfies $p(x_i) = y_i$ for all $i = 0, 1, \dots, n$. The polynomial $p(x)$ is said to interpolate the given data points. Interpolation can be performed with functions other than polynomials (although these are most common), with important cases being rational functions, trigonometric polynomials, and spline functions (made by connecting several polynomial functions at their endpoints—they are commonly used in statistics and computer graphics).

Interpolation has a number of applications. If a function $f(x)$ is known only at a discrete set of data points $x_0, \dots, x_n$, with $y_i = f(x_i)$, then interpolation can be used to extend the definition to nearby points x . If n is at all large, spline functions are generally preferable to simple polynomials[16].

Most numerical methods for the approximation of integrals and derivatives of a given function $f(x)$ are based on interpolation. For example, begin by constructing an interpolating function $p(x)$, often a polynomial, that approximates $f(x)$, and then integrate or differentiate $p(x)$ to approximate the corresponding integral or derivative of $f(x)$.

8-Solving differential and integral equations

Most mathematical models used in the natural sciences and engineering are based on ordinary differential equations, partial differential equations, and integral equations. Numerical methods for solving these equations are primarily of two types. The first type approximates the unknown function in the equation by a



simpler function, often a polynomial or piecewise polynomial (spline) function, chosen to closely follow the original equation. The finite element method discussed above is the best known approach of this type. The second type of numerical method approximates the equation of interest, usually by approximating the derivatives or integrals in the equation. The approximating equation has a solution at a discrete set of points, and this solution approximates that of the original equation. Such numerical procedures are often called finite difference methods. Most initial value problems for ordinary differential equations and partial differential equations are solved in this way. Numerical methods for solving differential and integral equations often involve both approximation theory and the solution of quite large linear and nonlinear systems of equations[17].

9-Effects of computer hardware

Almost all numerical computation is carried out on digital computers. The structure and properties of digital computers affect the structure of numerical algorithms, especially when solving large linear systems. First and foremost, the computer arithmetic must be understood. Historically, computer arithmetic varied greatly between different computer manufacturers, and this was a source of many problems when attempting to write software that could be easily ported between different computers. Variations were reduced significantly in 1985 with the development of the Institute for Electrical and Electronic Engineering (IEEE) standard for computer floating-point arithmetic. The IEEE standard has been adopted by all personal computers and workstations as well as most mainframe computers. For large-scale problems, especially in numerical linear algebra, it is important to know how the elements of an array A or a vector x are stored in memory. Knowing this can lead to much faster transfer of numbers from the memory into the arithmetic registers of the computer, thus leading to faster programs. A somewhat related topic is that of "pipelining." This is a widely used technique whereby the executions of computer operations are overlapped, leading to faster execution. Machines with the same basic clock speed can have very different program execution times due to differences in pipelining and differences in the way memory is accessed[18].

Most personal computers are sequential in their operation, but parallel computers are being used ever more widely in public and private research institutions. (See supercomputer.) Shared-memory parallel computers have several independent central processing units (CPUs) that all access the same computer memory, whereas distributed-memory parallel computers have separate memory for each CPU. Another form of parallelism is the use of pipelining of vector arithmetic operations. Numerical algorithms must be modified to run



most efficiently on whatever combination of methods a particular computer employs[19].

Conclusions

We have presented a 2D accurate and efficient numerical analysis of the reflector beam waveguides made of several PEC elliptic mirrors. This has been achieved with the aid of the SIE-MDS approach. The main feature of a beam waveguide is an ability to focus the beam and reproduce the near-field pattern after each reflector. As a simple figure of merit, we have studied the field amplitudes in the consecutive focal points of confocal elliptic reflectors under the variation of various parameters. Our analysis has shown that the ability of the chain of reflectors to guide the wave radiated by an aperture source strongly depends on the edge illumination of the first reflector and on the proper placement of the source. For in-focus source, the edge illumination of a 20 reflector should not be higher than -12 dB. Numerical synthesis is a much more complicated endeavor than analysis. We have tested the SIE-MDS technique as a full-wave engine for building a synthesis code based on a seminumerical gradient method. As we have found, if the objective function is the deviation of the near field from a given function on a certain open or closed contour, it is possible to derive a separate SIE for the gradient of the objective function and to solve it with the MDS as well. This speeds up the synthesis process and guarantees accuracy. Example of the synthesis of a shaped reflector has been given to demonstrate the splitting of the incident Gaussian-like beam into two high-intensity spots at prescribed locations.

References

1. Feature issue, "Diffractive optics applications," Appl. Opt. 34, 2399–2559 (1995).
2. J. M. Bendickson, E. M. Glytsis, and T. K. Gaylord, "Metallic surface-relief on-axis and off-axis focusing diffractive cylindrical mirrors," J. Opt. Soc. Am. A 16, 113–130 (1999).
3. J. M. Bendickson, E. N. Glytsis, and T. K. Gaylord, "Focusing diffractive cylindrical mirrors: rigorous evaluation of various design methods," J. Opt. Soc. Am. A 18, 1487–1494 (2001).
4. B. Z. Katsenelenbaum, "Transmission of millimetric waves by reflection from a series of focusing mirrors (Approximate theory of millimetric wave transmission, using reflections from row of focusing mirrors)," Radio Eng. Electron. Phys. 8, 1455–1460 (1963) [Radiotekh. Elektron. (Moscow) 8, 1516–1522 (1963)].



5. J. E. Degenford, M. D. Sirkis, and W. H. Steier, "The reflecting beam waveguide," IEEE Trans. Microwave Theory Tech. 12, 445–453 (1964).
6. B. Z. Katsenelenbaum, "Quasioptical methods of generation and transmission of millimeter waves," Sov. Phys. Usp. 7, 385–400 (1964) [Usp. Fiz. Nauk 83, 31–105 (1964)].
7. V. S. Averbakh, S. N. Vlasov, and V. I. Talanov, "Methods of mode selection in quasioptical systems," Radiophys. Quantum Electron. 10, 747–760 (1967) [Izv. Vyssh. Uchebn. Zaved. 10, 1333–1357 (1967)].
8. R. Tremblay and A. Boivin, "Concepts and techniques of microwave optics," Appl. Opt. 5, 249–278 (1966).
9. G. Goubau and P. D. Coleman, "Beam waveguides," in E. C. Okress, ed., Microwave Power Engineering (Academic, 1968), Vol. 1, pp. 228–255 (1968).
10. P. F. Goldsmith, Quasioptical Systems: Gaussian Beam, Quasioptical Propagation and Applications (IEEE, 1998).
11. A. Fernandez, K. M. Likin, P. Turullols, J. Teniente, R. Gonzalo, C. del Rio, J. Marti-Canales, M. Sorolla, R. Martin, "Quasioptical transmission lines for ECRH at TJ-II stellarator," Int. J. Infrared Millim. Waves 21, 1945–1957 (2000).
12. M. K. Thumm and W. Kasperek, "Passive high-power microwave components," IEEE Trans. Plasma Sci. 30, 755–786 (2002).
13. P. F. Goldsmith, "A quasi-optical feed system for radioastronomical observations at mm wavelength," Bell Syst. Tech. J. 56, 1483–1501 (1977).
14. L. B. Felsen, "Quasi-optic diffraction," in Proceedings of Symposium on Quasi-Optics (P. I. Brooklyn Press, 1964), pp. 1–40.
15. J. A. Murphy, "Distortion of a simple Gaussian beam on reflection from off-axis ellipsoidal mirrors," Int. J. Infrared Millim. Waves 8, 1165–1187 (1987).
16. L. Empacher and W. Kasperek, "Analysis of a multiplebeam waveguide for free-space transmission of microwaves," IEEE Trans. Antennas Propag. 49, 483–493 (2001).
17. T. Bondo and S. B. Sorensen, "Physical optics analysis of beam waveguides using auxiliary planes," IEEE Trans. Antennas Propag. 53, 1062–1068 (2005).



18. D. W. Prather, M. S. Mirotznik, and J. N. Mait, "Boundary integral methods applied to the analysis of diffractive optical elements," J. Opt. Soc. Am. A 14, 34–43 (1997).
19. J. M. Bendickson, E. N. Glytsis, and T. K. Gaylord, "Scalar integral diffraction methods: unification, accuracy, and comparison with a rigorous boundary element method with application to diffractive cylindrical lenses," J. Opt. Soc. Am. A 15, 1822–1837 (1998).
20. M. Testorf, "On the zero-thickness model of diffractive optical elements," J. Opt. Soc. Am. A 17, 1132–1133 (2000).
21. T. Oguzer, A. I. Nosich, and A. Altintas, "E-polarized beam scattering by an open cylindrical PEC strip having arbitrary "conical-section" profile," Microwave Opt. Technol. Lett. 31, 480–484 (2001). 22. A. A. Nosich and Y. V. Gandel, "Numerical analysis of quasi-optical multireflector antennas in 2-D with the method of discrete singularities," IEEE Trans. Antennas Propag. 55, 399–406 (2007).
23. J. Tsalamengas, "Exponentially converging Nystrom's methods for systems of SIEs with applications to open/ closed strip or slot-loaded 2-D structures," IEEE Trans. Antennas Propag. 54, 1549–1558 (2006).