ROYAUME DU MAROC
Université Mohammed V
- RABAT -
جامعة محمد الخامس
- الرباط -
Faculté des sciences
كلية العلوم
CENTRE D'ETUDES DOCTORALES - SCIENCES ET TECHNOLOGIES

N° d'ordre : 3509

THESE

En vue de l'obtention du : ***DOCTORAT***

Centre de Recherche *: CeReMAR*
Structure de Recherche *: Laboratoire Mathématiques Informatique et Applications – Sécurité de l'information (LabMiA-SI)*
Discipline *: Mathématiques Appliquées*
Spécialité *: Analyse Numérique*

Présentée et soutenue le 16/07/2021 par :

Safaa ELGHARBI

Méthodes Sinc exponentielles doubles, Méthodes de Lanczos par blocs et globale et applications

JURY

Souad EL BERNOUSSI	PES, Université Mohammed V - Rabat, Faculté des Sciences	Présidente
Hafida BENAZZA	PES, Université Mohammed V - Rabat, Faculté des Sciences	Rappoteure/Examinatrice
Youssef BENTALEB	PH, Université Ibn Tofail, ENSA - Kenitra	Rapporteur/Examinateur
Younes LOUARTASSI	PH, Université Mohammed V - Rabat, EST - Salé	Rapporteur/Examinateur
Bouchra ABOUZAID	PH, Université Chouaib Doukkali, ENSA - El Jadida	Examinatrice
Hassan SAFOUHI	PES, Université d'Alberta, Faculté Saint-Jean - Edmonton	Examinateur
Said EL HAJJI	Ex-PES, Université Mohammed V - Rabat, Faculté des Sciences	Invité
Mouna ESSAOUINI	PH, Université Chouaib Doukkali, Faculté des Sciences - El Jadida	Co-Directrice de Thèse
Mustapha ESGHIR	PH, Université Mohammed V - Rabat, Faculté des Sciences	Directeur de Thèse

Année Universitaire : 2020/2021

Remerciements

Ce travail a été effectué au laboratoire Mathématiques, Informatique et Applications - Sécurité de l'information, de la Faculté des Sciences de Rabat, tout d'abord sous la direction du professeur Said EL HAJJI et après sa retraite, ce travail a été achevé sous la direction des professeurs Mustapha ESGHIR et Mouna ESSAOUINI.

J'aimerais, remercier mon ancien directeur de thèse, monsieur Said EL HAJJI, ex-professeur d'enseignement supérieur à la Faculté des Sciences de Rabat, pour la confiance qu'il m'a accordée en acceptant d'encadrer ce travail doctoral avant de prendre sa retraite, ainsi que pour la bonne humeur et l'intérêt manifestés à l'égard de ma recherche qui m'ont permis de progresser dans cette phase délicate de mon apprentissage et recherche. Qu'il soit apprécié aussi pour ses qualités professionnelles et humaines qui m'ont facilité mon parcours doctoral.

Je souhaiterais exprimer ma gratitude à mon directeur de thèse, monsieur Mustapha ESGHIR, professeur habilité à la Faculté des Sciences de Rabat, pour ses multiples conseils et pour les heures qu'il a consacrées à diriger cette recherche. J'aimerais apprécier énormément ses qualités humaines d'écoute, de motivation et de compréhension tout au long de ces années de thèse doctorale, ainsi que sa disponibilité et son grand soutien à chaque fois que j'ai sollicité son aide.

Mes vifs remerciements vont également à ma co-directrice de thèse, madame Mouna ESSAOUINI, professeur habilitée à la Faculté des Sciences d'El Jadida, pour ses corrections et remarques pertinentes tout au long de ce travail de recherche. Je la remercie profondément pour ses multiples encouragements, notamment lors de mes communications aux congrès, ainsi que pour sa gentillesse, sa grande générosité et son accueil chaleureux à chaque fois que je me suis déplacée à El Jadida.

J'aimerais aussi adresser ma gratitude à madame Souad EL BERNOUSSI, professeur d'enseignement supérieur à la Faculté des Sciences de Rabat, pour avoir honoré cette soutenance de thèse par la présidence de son jury, ainsi que pour son assistance administrative en tant que responsable du laboratoire de Mathématiques, Informatique et Applications - Sécurité de l'Information.

Je tiens à remercier madame Hafida BENAZZA, professeur d'enseignement supérieur à la Faculté des Sciences de Rabat, pour l'intérêt qu'elle a manifesté à l'égard de cette recherche en s'engageant à être rapporteure et examinatrice.

Mes remerciements vont aussi à monsieur Youssef BENTALEB, professeur habilité à l'Ecole Nationale des Sciences Appliquées de Kénitra, pour avoir accepté d'être rapporteur et examinateur, et consacré son temps et effort pour juger ce travail.

Je voudrais remercier monsieur Younes LOUARTASSI, professeur habilitée à l'Ecole Supérieure de Technologie de Salé, d'avoir consacré son temps à examiner ce mémoire de thèse et d'avoir accepté d'être rapporteur et faire partie du jury.

J'aimerais également apprécier madame Bouchra ABOUZAID, professeur habilitée à l'Ecole Nationale des Sciences Appliquées d'El Jadida, pour l'intérêt qu'elle a porté à ce travail de recherche en acceptant la fonction d'examinatrice, aussi bien que pour son accueil enthousiaste à chaque fois que je l'ai rencontré.

Je saurais infiniment gré à monsieur Hassan SAFOUHI, professeur d'enseignement supérieur à l'université d'Alberta, pour ses vives critiques, ses suggestions utiles et surtout sa précieuse contribution qui a mené à achever ce travail de recherche.

Ma sincère reconnaissance va également à mes très chers parents pour leurs encouragements constants aussi bien que leur soutien affectif et matériel. Mes derniers remerciements vont à mes frères, ma famille, mes amis et mes collègues qui m'ont apporté leur soutien moral et intellectuel tout au long de cette période doctorale.

Abstract

The current thesis work is presented in two parts. The first part is devoted to the solution of the Schrödinger equation using Sinc numerical methods. The nonstationary version of this equation is treated in two dimensions using the double exponential Sinc collocation method (DESCM) and the double exponential Sinc-Galerkin method (DESGM). A comparison with the single exponential Sinc methods is made to illustrate the superiority of the DESinc methods. The resolution of the stationary Schrödinger equation using the DESCM and DESGM methods is also discussed in two- and three- dimensional spaces. Numerical experiments are reported to confirm the accuracy of the presented approaches for the Schrödinger equation with various separable and nonseparable potential functions. The matrices arising from the discretization of the equation using the DESinc methods are block centrosymmetric. This property is introduced as a two-dimensional extension of the well-known centrosymmetric property which helps to significantly reduce the computational time while calculating the eigenvalues of the system matrices.

The second part of this study is aimed at resolving large sparse linear systems with multiple right-hand sides using Lanczos based methods. The block nonsymmetric Lanczos method (Bl-NLM) as well as the global nonsymmetric Lanczos method (Gl-NLM) are introduced in new versions. A new expression of the solution is presented basing on some Schur complements identities and consequently some recursive formulas are occurred to give rise to new algorithms for implementing the block and global Lanczos methods. The proposed approaches are tested numerically and compared to other block and global Krylov subspace methods in order to illustrate their behavior and efficiency for some matrices. The combination of the global nonsymmetric Lanczos process and the Schur complements is also applied for the approximation of the transfer function. The global Lanczos method is used as a model reduction method to transform the original linear dynamical system into a smaller one and the Schur complements give new expressions of the approximate transfer function.

Keywords. Sinc collocation method, Sinc-Galerkin method, Double exponential transformations, Two- and three- dimensional Schrödinger equation, Block centrosymmetry. Large sparse linear systems with multiple right-hand sides, Block nonsymmetric Lanczos method, Global nonsymmetric Lanczos method, Schur complement, Transfer function.

Résumé

Ce travail de thèse est présenté en deux parties. La première partie est consacrée à la résolution de l'équation de Schrödinger par les méthodes numériques Sinc. La version non stationnaire de cette équation est traitée en deux dimensions en utilisant la méthode de collocation Sinc couplée avec des transformations exponentielles doubles (DESCM) et la méthode Sinc-Galerkin couplée avec des transformations exponentielles doubles (DESGM). Une comparaison avec les méthodes Sinc couplées avec des transformations exponentielles simples est faite pour illustrer la supériorité des méthodes DESCM et DESGM. La résolution de l'équation de Schrödinger stationnaire à l'aide des méthodes DESCM et DESGM est également discutée dans des espaces à deux et trois dimensions. Des expériences numériques sont rapportées pour confirmer l'exactitude des approches présentées pour l'équation de Schrödinger avec diverses fonctions potentielles séparables et non séparables. Les matrices issues de la discrétisation de l'équation à l'aide des méthodes DESinc sont centrosymétriques par blocs. Cette propriété est introduite comme une extension bidimensionnelle de la propriété centrosymétrique bien connue qui aide à réduire considérablement le temps d'exécution lors du calcul des valeurs propres des matrices du système.

La deuxième partie de cette étude vise à résoudre de grands systèmes linéaires creux avec plusieurs second membres en utilisant les méthodes de type Lanczos. La méthode de Lanczos non symétrique par blocs (Bl-NLM) ainsi que la méthode de Lanczos non symétrique globale (Gl-NLM) sont introduites dans de nouvelles versions. Une nouvelle expression de la solution est présentée en se basant sur certaines identités de compléments de Schur et, par conséquent, certaines formules récursives sont apparues pour donner naissance à de nouveaux algorithmes pour l'implémentation des méthodes Bl-NLM et Gl-NLM. Les approches proposées sont testées numériquement et comparées à d'autres méthodes de Krylov par blocs et globales afin d'illustrer leur comportement et leur efficacité pour certaines matrices. La combinaison du processus de Lanczos non symétrique global et des compléments de Schur est également appliquée pour l'approximation de la fonction de transfert. La méthode de Lanczos globale est utilisée comme méthode de réduction de modèle pour transformer le système dynamique linéaire en un système plus petit et les compléments de Schur donnent de nouvelles expressions de la fonction de transfert approchée.

Mots-clés. Méthode de collocation Sinc, Méthode Sinc-Galerkin, Transformations exponentielles doubles, Equation de Schrödinger en deux et trois dimensions, Centrosymmetrie par blocs. Grands systèmes linéaires denses avec plusieurs second membres, Méthode de Lanczos non symétrique par blocs, Méthode de Lanczos non symétrique globale, Complément de Schur, Fonction de transfert.

Résumé Détaillé

Les méthodes Sinc comblent un vide en mathématiques d'une manière similaire que les polynômes, les fonctions splines et les polynômes de Fourier. Il est clair qu'on a pu longtemps se débrouiller sans ces méthodes, mais leur présence a enrichi le domaine de l'analyse numérique. L'outil de base des approximations Sinc est la fonction cardinale qui représente un développement de la fonction à approximer. Les identités basées sur la fonction cardinale fournissent des approximations pour presque toutes les opérations de calcul. La fonction cardinale joue le même rôle pour les méthodes sinc que les polynômes pour les méthodes numériques classiques. Comme pour les polynômes, de nombreuses opérations sur la fonction cardinale peuvent être effectuées explicitement et chacune d'elles donne lieu à des méthodes explicites d'approximation, telles que la régle des trapèzes, la transformée de Fourier rapide et bien d'autres techniques faciles à déduire du développement cardinal.

La plupart des processus d'approximation numérique tels que les méthodes de quadrature, les méthodes des éléments finis et les méthodes des différences finies, sont basés sur des relations satisfaites par des polynômes. La simplicité et la diversité de ces identités ont permis la construction d'un grand nombre de procédures qui ont prouvé leur efficacité dans des voisinnages où la fonction à aborder est suffisamment régulière. Cependant, au voisinage des singularités, ces procédures sont généralement moins précises. Ceci est tout à fait compréhensible puisque chaque méthode existante excelle pour une classe particulière de problèmes. Par exemple, les fonctions polynomiales ont prouvé leur efficacité dans l'approximation de fonctions lisses qui ne souffrent pas de singularités tandis que les polynômes de Fourier excellent dans l'approximation de fonctions à la fois lisses et périodiques sur la droite réelle. Quant aux fonctions Sinc, elles ont montré leur talent dans l'approximation des fonctions analytiques et dans la résolution des problèmes avec des singularités définies sur des domaines infinis ou semi-infinis.

Au cours des dernières décennies, une large collection de méthodes numériques basées sur l'approximation Sinc a vu le jour, appelées méthodes numériques Sinc. Ces derniers, en tant qu'outils d'approximation, couvrent l'approximation de fonctions, de dérivées, d'intégrales définies et indéfinies et de convolutions définies et indéfinies. Ils couvrent également l'approximation des transformées de Fourier, des transformées de Laplace et des transformées de Hilbert, ainsi que l'approximation de la solution des équations différentielles ordinaires aux valeurs initiales et aux limites, des équations aux dérivées partielles et des équations intégrales.

La première partie de ce travail porte sur l'utilisation des méthodes numériques Sinc exponentielles doubles pour la résoltuion de l'équation de Schrödinger. La version non stationnaire de cette dernière est étudiée en deux dimensions en utilisant la méthode de collocation Sinc couplée avec des transformations exponentielles doubles (DESCM) et la méthode Sinc-Galerkin couplée avec des transformations exponentielles doubles (DESGM). On compare avec les méthodes Sinc couplées avec des transformations exponentielles simples pour illustrer l'exactitude des méthodes DESCM et DESGM pour l'approximation de la fonction d'onde.

La résolution de l'équation de Schrödinger stationnaire à l'aide des méthodes DESCM et DESGM est également discutée dans des espaces à deux et trois dimensions. Des exemples numériques sont présentés pour confirmer la supériorité des approches présentées lorsqu'elles sont appliquées

pour le calcul de l'énergie de l'équation de Schrödinger. Dans le cas des fonctions potentielles séparables en deux dimensions, la résolution du grand système obtenu est transformée à celle de deux problèmes valeur propre plus petits. D'ailleurs, on profite de l'étude unidimensionnelle pour choisir les transformations adéquates aussi bien que les paramètres liés au comportement asymptotique de la fonction d'onde.

L'équation de Schrödinger avec des potentiels non séparables est également traitée numériquement. En tenant compte de quelques hypothèses particulières sur les fonctions potentielles et les transformations exponentielles doubles, on constate que les marices issues de la discrétisation de l'équation à l'aide des méthodes DESinc sont centrosymétriques par blocs. Cette propriété est introduite comme une extension bidimensionnelle de la propriété centrosymétrique bien connue qui aide à réduire considérablement le temps d'exécution lors du calcul des valeurs propres des matrices du système.

Les méthodes numériques mentionnées précédemment, lorsqu'elles sont appliquées pour la discrétisation de plusieurs équations différentielles multidimensionnelles, aboutissent à de grands systèmes linéaires de taille finie liée au nombre de points de discrétisation. La résolution de tels systèmes reste un des problèmes majeurs rencontrés en analyse numérique, ce qui nécessite de trouver les méthodes appropriées capables de réduire le nombre d'itérations, la complexité du calcul, le temps d'exécution et l'espace de stockage. Un choix approprié de telles méthodes est basé sur la structure de la matrice du système (creuse, pleine, symétrique, définie positive, etc.). On peut trouver deux grandes familles de méthodes de résolution : les méthodes directes et les méthodes itératives.

Les méthodes de résolution directe consistent à exprimer la matrice système sous forme d'un produit de matrices plus faciles à inverser. Pendant le processus de décomposition, l'utilisation de la structure de la matrice et la réorganisation soigneuse des équations et des inconnues sont nécessaires afin de contrôler le positionnement des entrées non nulles dans le cas de matrices creuses. Les méthodes directes ont souvent été les plus utilisées dans les applications réelles grâce à leur fiabilité et leur comportement anticipé. Ils conviennent aux systèmes denses de taille modérée, plus précisément, lorsque l'espace mémoire est suffisant pour le stockage de l'ensemble de la matrice de coefficients. Cependant, en raison des exigences de calcul lorsqu'il s'agit de systèmes de grande taille, les méthodes directes sont donc considérés comme inefficaces et les méthodes itératives sont davantage préférées en raison de leur efficacité et de leur facilité de mise en oeuvre.

L'utilisation des méthodes itératives pour la résolution des grands systèmes linéaires creux a acquis de la popularité dans plusieurs domaines du calcul scientifique. Ces méthodes consistent à générer une suite de solutions approximatives qui converge vers la solution exacte. Pour de telles méthodes, seule la multiplication matrice-vecteur est utilisée lorsque la matrice du système est impliquée, et cette propriété rend ces méthodes encore plus intéressantes.

Les méthodes du sous-espace de Krylov sont considérées comme l'une des techniques itératives les plus populaires et les plus pertinentes pour la résolution des grands systèmes linéaires. Ces méthodes dépendent essentiellement de la projection du système original sur des sous-espaces de Krylov. Plus précisément, ces techniques visent à produire une base du sous-espace de Krylov et à trouver une solution approximative du problème dans ce sous-espace. Par conséquent, la matrice du système d'origine est approchée par une matrice beaucoup plus petite et structurée.

Les méthodes du sous-espace de Krylov comprennent plusieurs méthodes connues telles que : la méthode du gradient conjugué (CG), la méthode du gradient BiConjugué (BiCG), la méthode stabilisée du gradient BiConjugué (Bi-CGStab), la méthode des résidus minimals généralisés (GMRES) et la méthode de Lanczos. D'autre part, afin de résoudre des systèmes linéaires avec la même matrice de coefficients et des second membres différents, il y a eu des généralisations pour donner de nouvelles versions des méthodes classiques du sous-espace de krylov. On peut trouver les méthodes par blocs telles que la méthode de Lanczos par blocs, la méthode BiCG par blocs (Bl-BiCG), la méthode BiCGStab par blocs (Bl-BiCGStab) et la méthode GMRES par blocs (Bl-GMRES), et les méthodes globales telles que la méthode de Lanczos globale, la méthode BiCG globale (Gl-BiCG), la méthode BiCGStab globale (Gl-BiCGStab) et la méthode GMRES globale (Gl-GMRES).

Dans la deuxième partie de cette étude, on tente de résoudre de grands systèmes linéaires avec la même matrice creuse et plusieurs second membres en utilisant les méthodes de type Lanczos. Tel système survient dans plusieurs problèmes dans différents domaines. Les méthodes de Lanczos non symétrique par blocs (Bl-NLM) et de Lanczos non symétrique globale (Gl-NLM) sont introduites dans de nouvelles versions. On présente de nouvelles expressions de la solution en se basant sur certaines identités du complément de Schur et sur les processus de Lanczos non symétrique par blocs et de Lanczos non symétrique globale. Par conséquent, on développe certaines expressions récursives pour donner lieu à de nouveaux algorithmes pour l'implémentation des méthodes Bl-NLM et Gl-NLM.

Les approches proposées sont testées numériquement et comparées à d'autres méthodes de Krylov par blocs et globales afin d'illustrer leur comportement et leur efficacité pour certaines matrices. La combinaison du processus de Lanczos non symétrique global et du complément de Schur, précisement l'identité de Sylvester matricielle, est également appliquée pour l'approximation de la fonction de transfert. La méthode de Lanczos globale est utilisée comme méthode de réduction de modèle pour transformer le système dynamique linéaire en un système plus petit et le complément de Schur sert à donner de nouvelles expressions de la fonction de transfert approchée. Appliquer cette version directe de la méthode Gl-NLM permet de réduire le temps d'exécution ainsi que l'espace de stockage nécessaire pour approcher la fonction de transfer.

Publications

List of Publications

- **S. Elgharbi**, M. Essaouini, B. Abouzaid, S. El Hajji and H. Safouhi, Double exponential Sinc numerical methods for the two-dimensional time-independent Schrödinger equation, Molecular Physics, e1909162, 2021.
- **S. Elgharbi**, M. Esghir, O. Ibrihich, B. Abouzaid, M. Essaouini and S. El Hajji, Direct global Lanczos method for large linear systems with multiple right-hand sides, Afrika Matematika, 31(1): 57-69, 2020.
- **S. Elgharbi**, M. Esghir, O. Ibrihich, A. Abarda, S. El Hajji and S. Elbernoussi, Grey-Markov model for the prediction of the electricity production and consumption, Lecture Notes in Networks and Systems, 81: 206-219, 2020.
- M. Esghir, O. Ibrihich, **S. Elgharbi**, M. Essaouini and S. El Hajji, Solving large linear systems with multiple right-hand sides, Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017, 1-6, 2018.
- **S. Elgharbi**, M. Esghir and S. El Hajji, The global Lanczos method for the approximation of the transfer function, submitted 2021.

List of Talks

- **S. Elgharbi**, M. Essaouini, B. Abouzaid, S. El Hajji and H. Safouhi, Numerical method for solving the two-dimensional Schrödinger equation with irregular singularities. The International Conference: Mathematical Modeling with Applications (M2A19), April 1 - 4, 2019, Rabat, Morocco.
- **S. Elgharbi**, M. Esghir, O. Ibrihich, A. Abarda, S. El Hajji and S. Elbernoussi, Grey-Markov model for the prediction of the electricity production and consumption. The International Conference on Big Data and Business Intelligence (BDBI'19), April 29 - May 2, 2019, Leuven, Belgium.
- **S. Elgharbi**, M. Essaouini, B. Abouzaid, S. El Hajji and H. Safouhi, Computation of energy eigenvalues of the two-dimensional Schrödinger equation using the double exponential Sinc collocation method. International Conference on Advances in Energy Technologies, Environmental Engineering and Materials Science (AETEEMS 2018), December 13 - 14, 2018, El Jadida, Morocco.
- M. Esghir, O. Ibrihich, **S. Elgharbi**, M. Essaouini and S. El Hajji, Solving large linear systems with multiple right-hand sides. International Conference on Engineering and Technology (ICET'2017) by IARES, Antalya, Turkey, 2017.
- M. Esghir, O. Ibrihich, **S. Elgharbi**, M. Essaouini and S. El Hajji, Block Lanczos methods. 9^{th} Pan African Congress of Mathematicians, July 3 - 7, 2017, Rabat, Morocco.

List of Tables

List of Figures

Contents

General Introduction

Sinc methods fill a mathematical gap in a similar way as polynomials, spline functions and Fourier polynomials. It is clear that we could manage without these methods for a long time, but their presence has enriched the field of approximations and computer science. The basic tool of Sinc approximations is the cardinal function which represents an expansion of the function to be approximated. The identities based on the cardinal function provide approximations for almost all computation operations. The cardinal function plays the same role for Sinc methods as polynomials for classical numerical methods. As for polynomials, many operations on the cardinal function can be performed explicitly and each of them gives rise to explicit approximation methods such as the trapezoidal rule, the fast Fourier transform and several other techniques that are easy to deduce from the cardinal expansion.

Most of the numerical approximation processes such as: quadrature methods, finite element methods and finite difference methods are based on relations satisfied by polynomials. The simplicity and diversity of these identities have made possible the construction of a large number of procedures that have proven their effectiveness in neighborhoods where the function to be approached is sufficiently smooth. However, in the vicinity of singularities, these procedures are generally less accurate. This is quite understandable since each existing method excels for a particular class of problems. For example, polynomial functions have proved their efficiency in the approximation of smooth functions that do not suffer from singularities while Fourier polynomials excel in the approximation of functions which are both smooth and periodic on the entire real line. As for the Sinc functions, they have shown their talent in the approximation of analytical functions and while solving problems with singularities defined on infinite or semi-infinite domains.

Over the past decades, there have been emerged a wide collection of numerical methods based on Sinc approximation that are referred to as Sinc numerical methods. These latter, as approximation tools, cover the approximation of functions, derivatives, definite and indefinite integrals and definite and indefinite convolutions. They also cover the approximation of Fourier transforms, Laplace transforms and Hilbert transforms, as well as the approximation of the solution of initial and boundary ordinary differential equations, partial differential equations and integral equations.

The numerical methods mentioned previously, while applied for the discretization of various multidimensional differential equations, result in large linear systems of finite size related to the number of the gridpoints. Solving such systems remains a severe issue in numerical analysis. Therefore, it requires finding the appropriate methods taking into account the reduction of the number of iterations, the calculus complexity, the execution time and the storage space. An appropriate choice of such methods is based on the structure of the system matrix (sparse, full, symmetric,

positive definite, etc.). One can find two main families of resolution methods: direct methods and iterative methods.

Direct solution methods consist in expressing the system matrix as a product of matrices that are easier to invert. During the decomposition process, the use of the matrix data structure and the careful reorder of the equations and the unknowns is needed in order to control the fill-in of the nonzero entries in the case of sparse matrices. Direct methods were often recommended in real applications thanks to their reliability and anticipated behavior. They are suitable for dense systems of moderate size, more precisely, when the memory space is sufficient for the storage of the whole coefficient matrix. However, due to the computational requirements when dealing with large systems, direct solvers are thus considered inefficient and iterative methods are more preferred because of their effectiveness and ease of implementation.

The utilization of iterative methods for the solution of large sparse linear systems has gained popularity in several fields of scientific computing. These methods consist in generating a sequence of approximate solutions that converges to the solution at a satisfactory rate. For such methods, only the matrix-vector multiplication is used when the matrix of the system is involved, and this property makes these methods even more interesting.

The Krylov subspace methods are regarded to be among the most popular and relevant iterative techniques for the resolution of large linear systems. These methods depend basically on projecting the original system onto Krylov subspaces. More specifically, these techniques aim to produce a basis of the Krylov subspace and find an approximate solution of the problem in this subspace. Consequently, the original system matrix is approximated by a much smaller and structured matrix. The Krylov subspace methods include various known methods such as: Conjugate Gradient method (CG), BiConjugate Gradient method (BiCG), BiConjugate Gradient Stabilized method (Bi-CGStab), Generalized Minimum Residual method (GMRES) and Lanczos method. On the other hand, in order to solve linear systems with the same coefficient matrix and different right-hand sides, there have been generalizations to give new versions of the classical krylov subspace methods. One can find the block solvers such that Bl-Lanczos method, Bl-BiCG method, Bl-BiCGStab method and Bl-GMRES method, and the global solvers such as Gl-Lanczos method, Gl-BiCG method, Gl-BiCGStab method and Gl-GMRES method.

The current thesis is divided into two parts. The first part addresses the use of double exponential Sinc numerical methods for the Schrödinger equation. It includes three chapters: Chapter 2 studies the numerical resolution of the time-dependent Schrödinger equation in a two dimensional space using double exponential Sinc collocation and Sinc-Galerkin methods. Chapter 3 treats the two- and three- dimensional time-independent Schrödinger equation by applying double exponential Sinc methods for separable potential functions. The nonseparable potentials are considered in Chapter 4 in which the block centrosymmetry is introduced as an extension of the centrosymmetric property in two dimensions. The last three chapters are contained in the second part of this thesis in which we attempt to use some identities of the Schur complement to establish new versions of the block nonsymmetric Lanczos method and the global nonsymmetric Lanczos method, by developing new recursive expressions for the approximate solution. Chapter 5 and Chapter 6 present new variants of the block and global Lanczos methods for solving large sparse linear systems with the same coefficient matrix and different right-hand sides. Chapter 7 uses the incorporation of the global Lanczos method and the Schur complements in order to approximate the transfer function.

All along this thesis, numerical examples and comparisons have been shared to confirm the reliability and efficiency of the proposed approaches.

Chapter 1

General Preliminaries

In this chapter, we present some introductory material [1, 2] about the Sinc numerical methods. We share a brief insight into the Sinc function and some of its properties, and we provide the interpolation and the quadrature rules of the Sinc methods. This chapter also reviews the resolution of linear systems using the Lanczos method based on the Lanczos biorthogonalization. It recalls the definition and some of the properties of the Schur complement. The latter has several applications in applied mathematics and matrix theory [3, 4, 5]. Some products definitions and properties are also recalled to be helpful in upcoming chapters.

1.1 Sinc Definitions and properties

Definition 1.1.1 *The function defined on the whole real line by:*

$$sinc(x) = \begin{cases} \dfrac{\sin(\pi x)}{\pi x} & for \quad x \neq 0 \\ 1 & for \quad x = 0. \end{cases} \tag{1.1.1}$$

is called the normalized sinc function.

Definition 1.1.2 *For $h > 0$ and $k \in \mathbb{Z}$, the Sinc function $S(k, h)$ is defined by:*

$$S(k, h)(x) = sinc\left(\frac{x - kh}{h}\right). \tag{1.1.2}$$

Sinc functions satisfy the discrete orthogonality property, that is to say that for all $k, l \in \mathbb{Z}$:

$$S(k, h)(lh) = \delta_{k,l}, \tag{1.1.3}$$

where $\delta_{k,l}$ is the Kronecker delta function.

Theorem 1.1.3 *Let $\delta_{jk}^{(n)}$ for $n = 0, 1, 2$ stand for the n-th Sinc differentiation matrix which is given*

by:

$$\delta_{jk}^{(0)} \;=\; S(j,h)(x)\Big|_{x=kh} = \begin{cases} 1 & for \quad j=k \\ 0 & for \quad j\neq k \end{cases} \tag{1.1.4}$$

$$\delta_{jk}^{(1)} \;=\; h\frac{d}{dx}S(j,h)(x)\Big|_{x=kh} = \begin{cases} 0 & for \quad j=k \\ \dfrac{(-1)^{k-j}}{k-j} & for \quad j\neq k \end{cases} \tag{1.1.5}$$

$$\delta_{jk}^{(2)} \;=\; h^2\frac{d^2}{dx^2}S(j,h)(x)\Big|_{x=kh} = \begin{cases} \dfrac{-\pi^2}{3} & for \quad j=k \\ \dfrac{-2(-1)^{k-j}}{(k-j)^2} & for \quad j\neq k. \end{cases} \tag{1.1.6}$$

The following definition presents the so-called Whittaker cardinal expansion or cardinal function, which is an infinite series involving the Sinc basis functions.

Definition 1.1.4 *For a given function f defined on the whole real line $\mathbb{R}$, we define the Whittaker cardinal expansion of f by the following series:*

$$C(f,h) = \sum_{k=-\infty}^{\infty} f(kh)\, S(k,h)(x), \tag{1.1.7}$$

where $h>0$. The truncated cardinal expansion of f is defined by:

$$C_{M,N}(f,h)(x) = \sum_{k=-M}^{N} f(kh)\, S(k,h)(x), \tag{1.1.8}$$

where M and N are integers.

The following definition presents a class of functions that are successfully approximated by Sinc expansions.

Definition 1.1.5 *Let $d>0$ and let D_d denote the strip of width $2d$ about the real axis:*

$$D_d \;=\; \{z\in\mathbb{C}:\; |\, Im(z)\,| <\; d\}. \tag{1.1.9}$$

For $\epsilon\in(0,1)$, let $D_d(\epsilon)$ denote the rectangle in the complex plane:

$$D_d(\epsilon) \;=\; \{z\in\mathbb{C}:\; |\, Re(z)\,| <\; 1/\epsilon,\;\; |\, Im(z)\,| <\; d(1-\epsilon)\}. \tag{1.1.10}$$

Let $\boldsymbol{B}_p(D_d)$ denote the family of all functions g that are analytic in D_d, such that:

$$\int_{-d}^{d} |g(x+iy)|dy \to 0 \quad as \quad x\to\pm\infty \qquad and \qquad N_p(g,D_d) = \lim_{\epsilon\to 0}\left(\int_{\partial D_d(\epsilon)} |g(z)|^p|dz|\right)^{1/p} < \infty. \tag{1.1.11}$$

For $p=1$, let $N(g,D_d)\equiv N_1(g,D_d)$ and $\boldsymbol{B}(D_d)\equiv\boldsymbol{B}_1(D_d)$.

In several problems arising in applied mathematics, the whole real line is not the natural domain. In order to apply Sinc methods to problems given on general intervals, the processes can be adapted with the aid of appropriate selected conformal mappings. The basis functions on an interval $I \subset \mathbb{R}$ are then given by:

$$S_k(x) \equiv S(k,h) \circ \phi(x), \qquad k \in \mathbb{Z} \tag{1.1.12}$$

and the grid points $x_k \in I$ are:

$$x_k = \Phi(kh) = \phi^{-1}(kh). \tag{1.1.13}$$

where ϕ is a one-to-one transformation which maps I onto $\mathbb{R}$.

To implement the single exponential (SE) or the double exponential (DE) Sinc numerical methods, an adroit choice of the conformal mappings needs to be made to force the solution to decay single exponentially or double exponentially. The table below gives the selected transformations.

Interval	SE-transformation	DE-transformation
(a,b)	$\left(\frac{b-a}{2}\right)\tanh\left(\frac{t}{2}\right)+\left(\frac{b+a}{2}\right)$	$\left(\frac{b-a}{2}\right)\tanh\left(\frac{\pi}{2}\sinh(t)\right)+\left(\frac{b+a}{2}\right)$
$(0,\infty)$	$\exp(t)$	$\exp(\sinh(t))$
$(-\infty,\infty)$	$\sinh(t)$	$\sinh(\sinh(t))$

Table 1.1: Variable transformations

1.2 Convergence of Sinc methods

The following theorems show the convergence rate of the interpolation and the quadrature rules of the single exponential and the double exponential Sinc numerical methods.

Theorem 1.2.1 *Let $\phi' f \in \boldsymbol{B}(D_d)$, with $d > 0$. If there exist positive constants α_1, α_2, C so that:*

$$| f(x) | \leq C \begin{cases} \exp(-\alpha_1|\phi(x)|), & x \in \Gamma_1 \equiv \{x \in \Gamma : \phi(x) \in (-\infty, 0)\}, \\ \exp(-\alpha_2|\phi(x)|), & x \in \Gamma_2 \equiv \{x \in \Gamma : \phi(x) \in [0, \infty)\}, \end{cases} \tag{1.2.1}$$

where $\Gamma = \{\Phi(t) \in D_d : -\infty < t < \infty\}$, and if the selections $N = [\frac{\alpha_1}{\alpha_2}M + 1]$ and $h = \left(\frac{\pi d}{\alpha_1 M}\right)^{1/2} \leq \frac{2\pi d}{ln(2)}$ are made, then for all $x \in \Gamma$,

$$\left| f(x) - \sum_{k=-M}^{N} f(x_k) S(k,h) \circ \phi(x) \right| \leq K M^{1/2} \exp\left(-(\pi d \alpha_1 M)^{1/2}\right), \tag{1.2.2}$$

where K is a constant depending on f, d, ϕ, D_d.

Theorem 1.2.2 *[6, 7] Suppose there are positive constants β_1, β_2, γ_1, γ_2, C so that:*

$$|f(x)| \leq C \begin{cases} \exp(-\beta_1 exp(\gamma_1 |\phi(x)|)), & for \ x \in \Gamma_1, \\ \exp(-\beta_2 exp(\gamma_2 |\phi(x)|)), & for \ x \in \Gamma_2. \end{cases} \tag{1.2.3}$$

Let $\phi' f \in \boldsymbol{B}(D_d)$ with $d \leq \dfrac{\pi}{2\gamma}$ where $\gamma = \max\{\gamma_1, \gamma_2\}$ and make the selection $h = \dfrac{\log(\pi d\gamma n/\beta)}{\gamma n}$, where β and n are given in [6]. Then for all $x \in \Gamma$,

$$\left| f(x) - \sum_{k=-M}^{N} f(x_k) S(k,h) \circ \phi(x) \right| \leq K \exp\left(-\frac{\pi d\gamma n}{\log(\pi d\gamma n/\beta)} \right), \tag{1.2.4}$$

where K is a constant that depends on f, d, ϕ, D_d.

Theorem 1.2.3 *Let the function $\frac{f}{\phi'}$ verify the relation* (1.2.1) *and select N as in Theorem 1.2.1 and $h = \left(\frac{2\pi d}{\alpha_1 M}\right)^{1/2}$, then we have:*

$$\left| \int_\Gamma f(x)dx - h \sum_{k=-M}^{N} \frac{f(x_k)}{\phi'(x_k)} \right| \leq K \exp\left(-(2\pi d\alpha_1 M)^{1/2} \right), \tag{1.2.5}$$

where K is a constant depending on f, d, ϕ, D_d.

Theorem 1.2.4 *[6, 7] Let the function $\dfrac{f}{\phi'}$ verify the relation* (1.2.3) *and make the selections γ, n and β as in Theorem 1.2.2 and $h = \dfrac{\log(2\pi d\gamma n/\beta)}{\gamma n}$, then we have:*

$$\left| \int_\Gamma f(x)dx - h \sum_{k=-M}^{N} \frac{f(x_k)}{\phi'(x_k)} \right| \leq K \exp\left(-\frac{2\pi d\gamma n}{\log(\pi d\gamma n/\beta)} \right), \tag{1.2.6}$$

where K is a constant depending on f, d, ϕ, D_d.

1.3 The Lanczos method

In this section, we aim to apply the Lanczos method for the resolution of the linear system:

$$Ax = b, \tag{1.3.1}$$

with A is an n by n real nonsymmetric and nonsingular matrix, b is a real n-vector and x is the n-vector of unknowns.

1.3.1 Lanczos biorthogonalization procedure

Given an $n \times n$ real nonsymmetric and nonsingular matrix A and initial vectors v_1, w_1 such that $(v_1, w_1) = 1$. The Lanczos biorthogonalization process consists in constructing two bi-orthogonal bases $\{v_1, v_2, \ldots, v_m\}$ and $\{w_1, w_2, \ldots, w_m\}$ of the two Krylov subspaces:

$$K_m(A, v_1) = \text{span}\left\{v_1, Av_1, \ldots, A^{m-1}v_1\right\} \tag{1.3.2}$$

and

$$K_m(A^\top, w_1) = \text{span}\left\{w_1, A^\top w_1, \ldots, \left(A^\top\right)^{m-1} w_1\right\}, \tag{1.3.3}$$

respectively, such that $(v_i, w_j) = \delta_{ij},\ i, j = 1, \ldots, m$. The algorithm is given as follows:

Algorithm 1 Lanczos biorthogonalization process *[8]*

1 Choose two vectors v_1 *and* w_1 *in* $\mathbb{R}^n$ *such that* $(v_1, w_1) = 1$*;*

2 Set $\beta_1 = \delta_1 = 0$ *and* $w_0 = v_0 = 0$*;*

3 For $j = 1, 2, \ldots, m$ *Do:*

- $\alpha_j = (Av_j, w_j)$
- $\tilde{v}_{j+1} = Av_j - \alpha_j v_j - \beta_j v_{j-1}$
- $\tilde{w}_{j+1} = A^\top w_j - \alpha_j w_j - \delta_j w_{j-1}$
- $\delta_{j+1} = \left|\,(\tilde{v}_{j+1}, \tilde{w}_{j+1})\,\right|^{1/2}$
- $\beta_{j+1} = \left|\,(\tilde{v}_{j+1}, \tilde{w}_{j+1})\,\right| / \delta_{j+1}$
- $v_{j+1} = \tilde{v}_{j+1} / \delta_{j+1}$
- $w_{j+1} = \tilde{w}_{j+1} / \beta_{j+1}$

End Do.

This algorithm may break down in the j-th iteration if $(\tilde{v}_{j+1}, \tilde{w}_{j+1}) = 0$.

Denote $V_m = [v_1, v_2, \ldots, v_m]$ and $W_m = [w_1, w_2, \ldots, w_m]$ two $n \times m$ matrices. Let T_m be the $m \times m$ tridiagonal matrix defined by:

$$T_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & 0 \\ \delta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \delta_{m-1} & \alpha_{m-1} & \beta_m \\ 0 & & & \delta_m & \alpha_m \end{pmatrix} \tag{1.3.4}$$

where α_i, β_i and δ_i are the scalars defined in Algorithm 1.

Proposition 1.3.1 *[8] Assume that the Algorithm 1 does not break down before the $m-th$ step. Then $\{v_1, v_2, \ldots, v_m\}$ and $\{w_1, w_2, \ldots, w_m\}$ form bi-orthogonal bases of the Krylov subspaces $K_m(A, v_1)$ and $K_m(A^\top, w_1)$, respectively. Then, we have the following relations:*

$$AV_m = V_m T_m + \delta_{m+1} v_{m+1} e_m^\top, \tag{1.3.5}$$

$$A^\top W_m = W_m T_m^\top + \beta_{m+1} w_{m+1} e_m^\top, \tag{1.3.6}$$

$$W_m^\top A V_m = T_m. \tag{1.3.7}$$

where $e_m = (0, \ldots, 0, 1)^\top \in \mathbb{R}^m$.

1.3.2 Lanczos algorithm for linear systems

Consider the linear system (1.3.1). Assume that an arbitrary initial guess to the solution is given by x_0 and let the corresponding residual be $r_0 = b - Ax_0$. Recall that the Lanczos method for solving the linear system (1.3.1) consists in determining an approximate solution x_m such that:

$$x_m - x_0 \in K_m(A, r_0) \tag{1.3.8}$$

$$r_m = b - Ax_m \perp K_m\left(A^\top, \tilde{r}_0\right) \tag{1.3.9}$$

where $\tilde{r}_0$ is a column vector chosen so that $(r_0, \tilde{r}_0) \neq 0$.

Let $\{v_1, v_2, \ldots, v_m\}$ and $\{w_1, w_2, \ldots, w_m\}$ the sets of vectors defined by Algorithm 1 and which span the Krylov subspaces $K_m(A, r_0)$ and $K_m(A^\top, \tilde{r}_0)$, respectively, with the initialization $v_1 = \dfrac{r_0}{\|r_0\|_2}$ and w_1 satisfies $(v_1, w_1) = 1$.

The relation (1.3.8) is expressed as:

$$x_m = x_0 + V_m y_m, \tag{1.3.10}$$

where y_m is the vector of $\mathbb{R}^m$ obtained by the relation (1.3.9). That is:

$$(r_0 - AV_m y_m, w_i) = 0, \tag{1.3.11}$$

for $i = 1, \ldots, m$, which gives:

$$W_m^\top A V_m \, y_m = W_m^\top r_0, \tag{1.3.12}$$

$$W_m^\top r_0 = W_m^\top (\beta v_1) = \beta e_1, \tag{1.3.13}$$

where $\beta = \|r_0\|_2$ and $e_1 = (1, 0, \ldots, 0)^\top \in \mathbb{R}^m$. Consequently, the approximate solution takes the following form:

$$x_m = x_0 + V_m y_m, \tag{1.3.14}$$

with

$$y_m = T_m^{-1} (\beta e_1), \tag{1.3.15}$$

where T_m given by (1.3.7) is assumed to be nonsingular.

1.4 The Schur complement

This section recalls the definition of the Schur complement [9] and gives some of its properties. The notion of Schur complement was first introduced by Schur [10]. It is of great importance in numerical analysis as well as in linear algebra [4, 11, 12]. The Schur complements are used in later chapters to present new variants of the block and global Lanczos methods.

Definition 1.4.1 *Let M be a matrix partitioned into four blocks:*

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \tag{1.4.1}$$

where the submatrix D is assumed to be square and nonsingular. The Schur complement of D in M, denoted by (M/D), is defined by:

$$(M/D) = A - BD^{-1}C. \tag{1.4.2}$$

We can similarly define the Schur complement as follows:

$$(M/A) = D - CA^{-1}B \qquad \text{if } A \text{ is a nonsingular matrix,} \tag{1.4.3}$$
$$(M/B) = C - DB^{-1}A \qquad \text{if } B \text{ is a nonsingular matrix,} \tag{1.4.4}$$
$$(M/C) = B - AC^{-1}D \qquad \text{if } C \text{ is a nonsingular matrix.} \tag{1.4.5}$$

The following propositions show some properties of the Schur complement.

Proposition 1.4.2 *Assume that the submatrix D is nonsingular, then we have:*

$$\left(\begin{bmatrix} A & B \\ C & D \end{bmatrix}/D\right) = \left(\begin{bmatrix} D & C \\ B & A \end{bmatrix}/D\right) \tag{1.4.6}$$
$$= \left(\begin{bmatrix} C & D \\ A & B \end{bmatrix}/D\right) \tag{1.4.7}$$
$$= \left(\begin{bmatrix} B & A \\ D & C \end{bmatrix}/D\right). \tag{1.4.8}$$

Let M be the matrix defined by (1.4.1) and K the matrix partitioned as follows:

$$K = \begin{bmatrix} E & F & G \\ H & A & B \\ L & C & D \end{bmatrix}. \tag{1.4.9}$$

If the matrices A and M are nonsingular, then the following property holds:

Proposition 1.4.3 *The matrix Sylvester identity*

$$(K/M) = ((K/A)/(M/A)) \tag{1.4.10}$$

$$= \left(\begin{bmatrix} E & F \\ H & A \end{bmatrix} /A \right) - \left(\begin{bmatrix} F & G \\ A & B \end{bmatrix} /A \right) (M/A)^{-1} \left(\begin{bmatrix} H & A \\ L & C \end{bmatrix} /A \right). \tag{1.4.11}$$

1.5 The Kronecker product $\otimes$

The Kronecker product is defined for two matrices of arbitrary sizes.

Definition 1.5.1 *Let $A = (a_{ij}) \in \mathbb{R}^{m\times n}$ and $B = (b_{ij}) \in \mathbb{R}^{p\times q}$, the Kronecker product of A and B, denoted $A \otimes B$, is defined by:*

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{pmatrix} \in \mathbb{R}^{mp\times nq} \tag{1.5.1}$$

The following proposition gives some of the Kronecker product properties.

Proposition 1.5.2 *Let $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$. We have:*

- $(\alpha A) \otimes B = A \otimes (\alpha B) = \alpha(A \otimes B), \quad \alpha \in \mathbb{C}$,
- $(A \otimes B)^\top = A^\top \otimes B^\top$,
- *if $C \in \mathbb{R}^{m\times n}$, then $(A + C) \otimes B = A \otimes B + C \otimes B$,*
- *if $C \in \mathbb{R}^{p\times q}$, then $A \otimes (B + C) = A \otimes B + A \otimes C$,*
- *if $C \in \mathbb{R}^{r\times s}$, then $(A \otimes C) \otimes B = A \otimes (B \otimes C)$,*
- *if $C \in \mathbb{R}^{n\times s}$ and $D \in \mathbb{R}^{q\times t}$, then $(A \otimes B)(C \otimes D) = AC \otimes BD$,*
- *if $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{p\times p}$ are nonsingular, then $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$.*

1.6 The $\diamond$ product

Let X and Y be two n by s matrices. We consider the inner product:

$$< X, Y >_F = \operatorname{tr}\left(X^\top Y\right). \tag{1.6.1}$$

Let $\|.\|_F$ denote the Frobenius norm associated to this product.

Definition 1.6.1 *Let $\mathcal{V} = [V_1, V_2, \ldots, V_k]$ and $\mathcal{W} = [W_1, W_2, \ldots, W_k]$, where $V_i, W_i \in \mathbb{R}^{n\times s}$. The $\diamond$ product is defined by:*

$$\mathcal{V}^\top \diamond \mathcal{W} = \begin{pmatrix} < V_1, W_1 >_F & < V_1, W_2 >_F & \ldots & < V_1, W_l >_F \\ < V_2, W_1 >_F & < V_2, W_2 >_F & \ldots & < V_2, W_l >_F \\ \vdots & \vdots & \ddots & \vdots \\ < V_k, W_1 >_F & < V_k, W_2 >_F & \ldots & < V_k, W_l >_F \end{pmatrix}. \tag{1.6.2}$$

The product $\diamond$ verifies the following properties:

Proposition 1.6.2 *Let $A, B, C \in \mathbb{R}^{n\times ks}, D \in \mathbb{R}^{n\times n}, L \in \mathbb{R}^{k\times k}$ and $\alpha \in \mathbb{R}$. Then, we have:*

- $(A + B)^\top \diamond C = A^\top \diamond C + B^\top \diamond C$,
- $A^\top \diamond (B + C) = A^\top \diamond B + A^\top \diamond C$,
- $(\alpha A)^\top \diamond C = \alpha(A^\top \diamond C)$,
- $(A^\top \diamond B)^\top = B^\top \diamond A$,
- $(DA)^\top \diamond B = A^\top \diamond (D^\top B)$,
- *if* $A^\top \diamond A = I_k$ *and* $A(L \otimes I_s) = B$, *then* $A^\top \diamond B = L$,
- $A^\top \diamond (B(L \otimes I_s)) = (A^\top \diamond B)L$,
- $(A(L \otimes I_s))^\top \diamond B = L^\top \left(A^\top \diamond B\right)$.

Part I

Double exponential Sinc methods for the Schrödinger equation

Introduction

Over the last four decades, Sinc methods have been widely used to approximate functions over bounded and unbounded regions, derivatives, definite and indefinite integrals [1, 13, 2]. They have also been used to solve initial and boundary value ordinary differential equations, partial differential equations, and integral equations, particularly when the solutions have singularities, infinite domains, or boundary layers [14, 15, 7, 16, 17].

The Sinc-Galerkin method [1] depends on Galerkin's idea that the residual must be orthogonal to each Sinc basis function, and have been applied by many researchers, including Stenger [18] who used the method to solve boundary value problems, McArthur et al. [19] who introduced a numerical implementation of the method for second-order hyperbolic equations, and Lund [20] who proposed a symmetrization of the method for boundary value problems.

The Sinc collocation method [21, 22] consists in approximating the solution using Sinc basis functions and collocating the approximate solution at the Sinc grid points, which results in a matrix eigenvalue problem and gives an approximation of the eigenpair of the initial problem. The Sinc method has been applied to a number of numerical problems. For example, Carlson et al. [23] used the Sinc collocation method for initial value problems, Amore [24] applied a variation of the method for strong coupled problems, and Wu et al. [25] applied a boundary treatment when using the method to solve two-point boundary value problems.

Initially, the Sinc methods only provided approximations on the whole real line, but it has been shown that they also provide approximations over arbitrary intervals and contours by using appropriate variable transformations [26]. The choice of the transformation plays an important role in improving the convergence of the methods. Many authors [27, 28] have focused on single exponential Sinc methods, which combine Sinc methods and single exponential transformations and provide efficient results for a large class of problems with a convergence rate of $O(\exp(-\kappa\sqrt{m}))$ for some $\kappa > 0$, where m is the number of nodes used. The double exponential transformations came twenty years later following the single exponential transformations. Despite, they have known a big concern for a long time. The earliest seems to be due to Takahasi and Mori [29] in 1974 for numerically evaluating integrals. Years later, thanks to Sugihara [30], it turned out that the double exponential transformation can be paired with the Sinc expansion, resulting in a better convergence rate which is $O(\exp(-\kappa'\sqrt{m}/\log(m)))$ for some $\kappa' > 0$. Sugihara [31] also proved the optimality of this convergence rate realized by the double exponential transformations.

The double exponential Sinc collocation method (DESCM), which combines the double exponential transformations with the Sinc collocation method, has gained popularity due to its efficiency

and accuracy. It is qualified by its double exponential decaying and its rapid convergence rate which rarely make it face the instability problems. Safouhi's group have used the DESCM to deal with several one-dimensional problems. In particular, they have applied the method to compute the eigenvalues of the one-dimensional singular Sturm-liouville problems [6], the energy eigenvalues of the one-dimensional time-independent Schrödinger equation with anharmonic oscillators [32], the anharmonic Coulombic potentials [33] where a scaling factor was introduced to improve convergence and increase the stability of the approach, and the eigenvalues of Harmonic oscillators with a perturbation by a rational function [34]. Essaouini et al. [35] proceeded to use the DESCM for the anharmonic Coulombic potentials with irregular singularities.

The matrices stemming from the DESinc methods resolution of the one-dimensional Schrödinger equation are centrosymmetric. The latter property has been studied extensively long ago [36, 37]. Although, recent years has showed more interest in the properties of this type of matrices resulting from the resolution of linear equations and inverse eigenvalue problems using iterative methods [38, 39]. In one dimension [40], the authors took advantage of the symmetric property of the Sturm-Liouville operator and used it for the discretization of the differential equation. The DESCM has been applied to the Sturm-Liouville eigenvalue problems and using the eigenspectrum properties of symmetric centrosymmetric matrices, it has been proved that the resolution of the resulting generalized eigenvalue problem is equivalent to that of two smaller problems which gave more gain in efficiency and a substantially reduction in the computation complexity.

In this first part of this thesis, we aim to apply the double exponential Sinc collocation method and the double exponential Sinc-Galerkin method for the time-dependent Schrödinger equation in two dimensions and for the time-independent Schrödinger equation in two and three dimensions. We take advantage from the separable case which helps transform solving one large eigenvalue system into solving two or three smaller one-dimensional eigenvalue problems. We treat the nonseparable case where we use the centrosymmetric system matrix and expand its properties to a bidimensional space.

Chapter 2

Double exponential Sinc methods for the time-dependent Schrödinger equation

The current chapter provides a numerical solution of the time-dependent Schrödinger equation in a two dimensional space, for the wave function $\psi(x, y, t)$ with nonhomogeneous initial and boundary conditions:

$$-i\frac{\partial \psi}{\partial t} = \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} + V(x, y)\psi, \qquad (x, y) \in \Omega \subset \mathbb{R}^2,\ 0 < t \leq T, \tag{2.0.1}$$

$$\psi(x, y, 0) = f(x, y), \qquad (x, y) \in \Omega \subset \mathbb{R}^2, \tag{2.0.2}$$

$$\psi(x, y, t) = g(x, y, t), \qquad (x, y) \in \partial\Omega,\ 0 < t \leq T, \tag{2.0.3}$$

where i is the imaginary unit, V, f, g are smooth known functions, T is the final time and $\partial\Omega$ is the boundary of Ω. We select the domain $\Omega = (a, b) \times (a, b)$.

This form of the equation is the most general form which gives a description of systems evolving with time. It is used in physics, more especially in quantum mechanics and most of chemistry to deal with problems about the atomic structure of matter. Molecular dynamics [41], electromagnetics [42], diffraction optics [43] and collision dynamics [44] are some of the fields that are governed by such kind of an equation. Its resolution basically involves the computation of the function ψ which is a wave that describes the quantum aspects of the system.

Numerous methods have been used for the numerical solution of the time-dependent Schrödinger equation by various researchers. Dehghan et al. [45] proposed a numerical scheme to solve the equation using collocation points and approximating the solution using multiquadrics and thin plate splines radial basis functions. In another work [46], they applied a compact finite difference approximation of fourth-order for discretizing spatial derivatives and a boundary value method of fourth order for the time integration of the resulting linear system of ordinary differential equations. In [47], Sinc collocation and Sinc-Galerkin methods have been used for solving the two-dimensional Schrödinger equation with nonhomogeneous boundary conditions. An implicit semi-discrete higher order compact scheme, with an averaged time discretization, has been presented by Kalita et al.

[48] for the numerical solution of unsteady two-dimensional Schrödinger equation. The bi-conjugate gradient stabilized method have been employed for solving the complex algebraic system arising at every time level. Numerical schemes based on a Crank-Nicolson-type discretization of the equation were studied by Antoine et al. [49] using non-reflecting boundary conditions. The full discretization is realized by way of a standard finite element method to preserve the stability of the scheme. For the solution of the time dependent Schrödinger equation in its application to physical and chemical molecular phenomena, Kosloff [41] presented a Fourier method based on discretizing space and time on a grid to produce both spatial and time derivatives. The method conserves norm and energy, and preserves quantum mechanical commutation relations.

This chapter is aimed at applying the double exponential Sinc collocation method and the double exponential Sinc-Galerkin method for the resolution of the two-dimensional time-dependent Schrödinger equation. A numerical comparison between the double exponential and single exponential approaches is provided in the last section.

2.1 Double exponential Sinc collocation method

First of all, in order to deal with the time dependency, we discretize the equation (2.0.1) using the θ-weighted scheme:

$$-i\,\frac{\psi^{n+1}-\psi^{n}}{dt}=\theta\left(\psi_{xx}^{n+1}+\psi_{yy}^{n+1}+V(x,y)\psi^{n+1}\right)+(1-\theta)\left(\psi_{xx}^{n}+\psi_{yy}^{n}+V(x,y)\psi^{n}\right), \tag{2.1.1}$$

where $0\leq\theta\leq 1$, dt is the temporal step length with $t^{n+1}=t^n+dt$, ψ^n is the value of the unknown function ψ at time $t^n=n\,dt$ and ψ_{xx} and ψ_{yy} denote the second derivatives of ψ with respect to x and y, respectively. The approximate solution of the equation (2.0.1) at time t^n can be written as:

$$\psi_A(x,y,t^n)\equiv\psi_A^n(x,y)=\sum_{i=-M_x-1}^{N_x+1}\sum_{j=-M_y-1}^{N_y+1}u_{ij}^n\zeta_i(x)\zeta_j(y), \tag{2.1.2}$$

where $u_{i,j}^n$ are the coefficients that must be determined and the two linear functions are given by:

$$\zeta_i(x)=\begin{cases}\frac{b-x}{b-a}, & i=-M_x-1,\\ S_i(x), & i=-M_x,\ldots,N_x,\\ \frac{x-a}{b-a}, & i=N_x+1\end{cases}\qquad\text{and}\qquad\zeta_j(y)=\begin{cases}\frac{b-y}{b-a}, & j=-M_y-1,\\ S_j(y), & i=-M_y,\ldots,N_y,\\ \frac{y-a}{b-a}, & j=N_y+1.\end{cases} \tag{2.1.3}$$

where $S_i(x)=S(i,h_x)\circ\phi_x(x)$ and $S_j(y)=S(j,h_y)\circ\phi_y(y)$. Using the boundary conditions (2.0.2), the approximate solution becomes:

$$\psi_A(x,y,t^n)=\psi_s(x,y,t^n)+\psi_b(x,y,t^n), \tag{2.1.4}$$

where

$$\psi_s(x,y,t^n)=\sum_{i=-M_x}^{N_x}\sum_{j=-M_y}^{N_y}u_{ij}^nS_i(x)S_j(y), \tag{2.1.5}$$

$$\psi_b(x,y,t^n)=\chi(a,y,t^n)\,\zeta_{-Mx-1}(x)+\sigma(b,y,t^n)\,\zeta_{N_x+1}(x)+g(x,a,t^n)\,\zeta_{-My-1}(y)+g(y,b,t^n)\,\zeta_{N_y+1}(y), \tag{2.1.6}$$

with

$$\chi(a,y,t^n) = -g(a,a,t^n)\,\zeta_{-My-1}(y) + g(a,y,t^n) - g(a,b,t^n)\,\zeta_{N_y+1}(y), \tag{2.1.7}$$

$$\sigma(b,y,t^n) = -g(b,a,t^n)\,\zeta_{-My-1}(y) + g(b,y,t^n) - g(b,b,t^n)\,\zeta_{N_y+1}(y). \tag{2.1.8}$$

The Sinc collocation method consists in substituting the approximate solution into the Schrödinger equation and calculating the resulting system at the Sinc grid points:

$$(x_k, y_l) = \left(\phi_x^{-1}(kh_x), \phi_y^{-1}(lh_y)\right), \tag{2.1.9}$$

for $k = -M_x, \ldots, N_x$ and $l = -M_y, \ldots, N_y$. That is:

$$-i\,\frac{\psi_A^{n+1} - \psi_A^n}{dt} = \theta\,\left((\psi_A)_{xx}^{n+1} + (\psi_A)_{yy}^{n+1} + V(x_k,y_l)\,\psi_A^{n+1}\right) + (1-\theta)\,\left((\psi_A)_{xx}^{n} + (\psi_A)_{yy}^{n} + V(x_k,y_l)\,\psi_A^{n}\right). \tag{2.1.10}$$

Thus,

$$\begin{aligned} &-i\,\frac{\psi_s^{n+1} - \psi_s^n}{dt} - \theta\,\left((\psi_s)_{xx}^{n+1} + (\psi_s)_{yy}^{n+1} + V(x_k,y_l)\,\psi_s^{n+1}\right) = (1-\theta)\,\left((\psi_s)_{xx}^{n} + (\psi_s)_{yy}^{n} + V(x_k,y_l)\,\psi_s^{n}\right) \\ &+i\,\frac{\psi_b^{n+1} - \psi_b^n}{dt} + \theta\,\left((\psi_b)_{xx}^{n+1} + (\psi_b)_{yy}^{n+1} + V(x_k,y_l)\,\psi_b^{n+1}\right) + (1-\theta)\,\left((\psi_b)_{xx}^{n} + (\psi_b)_{yy}^{n} + V(x_k,y_l)\,\psi_b^{n}\right). \end{aligned} \tag{2.1.11}$$

Multiplying by dt, we obtain:

$$\begin{aligned} &-i\,\psi_s^{n+1}(x_k,y_l) - \theta\,dt\,\left((\psi_s)_{xx}^{n+1}(x_k,y_l) + (\psi_s)_{yy}^{n+1}(x_k,y_l) + V(x_k,y_l)\,\psi_s^{n+1}(x_k,y_l)\right) \\ &\quad = i\,\psi_b^{n+1}(x_k,y_l) + \theta\,dt\,\left((\psi_b)_{xx}^{n+1}(x_k,y_l) + (\psi_b)_{yy}^{n+1}(x_k,y_l) + V(x_k,y_l)\,\psi_b^{n+1}(x_k,y_l)\right) \\ &\qquad -i\,\psi_s^{n}(x_k,y_l) + (1-\theta)\,dt\,\left((\psi_s)_{xx}^{n}(x_k,y_l) + (\psi_s)_{yy}^{n}(x_k,y_l) + V(x_k,y_l)\,\psi_s^{n}(x_k,y_l)\right) \\ &\qquad\quad -i\,\psi_b^{n}(x_k,y_l) + (1-\theta)\,dt\,\left((\psi_b)_{xx}^{n}(x_k,y_l) + (\psi_b)_{yy}^{n}(x_k,y_l) + V(x_k,y_l)\,\psi_b^{n}(x_k,y_l)\right). \end{aligned} \tag{2.1.12}$$

It follows that:

$$\begin{aligned} &-\theta\,dt\sum_{i=-M_x}^{N_x}\sum_{j=-M_y}^{N_y} u_{ij}^{n+1}\left[S_i''(x_k)S_j(y_l) + S_i(x_k)S_j''(y_l) + V(x_k,y_l)S_i(x_k)S_j(y_l)\right] - i\,\psi_b(x_k,y_l,t^{n+1}) \\ &\quad -i\sum_{i=-M_x}^{N_x}\sum_{j=-M_y}^{N_y} u_{ij}^{n+1}S_i(x_k)S_j(y_l) = -i\sum_{i=-M_x}^{N_x}\sum_{j=-M_y}^{N_y} u_{ij}^{n}S_i(x_k)S_j(y_l) - i\,\psi_b(x_k,y_l,t^n) \\ &\qquad + (1-\theta)\,dt\sum_{i=-M_x}^{N_x}\sum_{j=-M_y}^{N_y} u_{ij}^{n}\left[S_i''(x_k)S_j(y_l) + S_i(x_k)S_j''(y_l) + V(x_k,y_l)S_i(x_k)S_j(y_l)\right] \\ &\qquad\quad + \theta\,dt\left[(\psi_b)_{xx}(x_k,y_l,t^{n+1}) + (\psi_b)_{yy}(x_k,y_l,t^{n+1}) + V(x_k,y_l)\psi_b(x_k,y_l,t^{n+1})\right] \\ &\qquad\qquad + (1-\theta)\,dt\left[(\psi_b)_{xx}(x_k,y_l,t^{n}) + (\psi_b)_{yy}(x_k,y_l,t^{n}) + V(x_k,y_l)\psi_b(x_k,y_l,t^{n})\right]. \end{aligned} \tag{2.1.13}$$

The Sinc function computation requires using the n-th derivatives $\delta_{ik}^{(n)},\ n=0,1,2$, of the Sinc function at the Sinc nodes. Thus, we get:

$$
\begin{aligned}
&-i\sum_{i=-M_x}^{N_x}\sum_{j=-M_y}^{N_y}u_{ij}^{n+1}\delta_{ik}^{(0)}\delta_{jl}^{(0)}-\theta\,dt\sum_{i=-M_x}^{N_x}\sum_{j=-M_y}^{N_y}u_{ij}^{n+1}\left[\left(\frac{1}{h_x}\phi_x''(x_k)\delta_{ik}^{(1)}+\frac{1}{h_x^2}\phi_x'(x_k)^2\delta_{ik}^{(2)}\right)\delta_{jl}^{(0)}\right.\\
&\left.+\left(\frac{1}{h_y}\phi_y''(y_l)\delta_{jl}^{(1)}+\frac{1}{h_y^2}\phi_y'(y_l)^2\delta_{jl}^{(2)}\right)\delta_{ik}^{(0)}+V(x_k,y_l)\delta_{ik}^{(0)}\delta_{jl}^{(0)}\right]=-i\sum_{i=-M_x}^{N_x}\sum_{j=-M_y}^{N_y}u_{ij}^{n}\delta_{ik}^{(0)}\delta_{jl}^{(0)}\\
&+i\,\psi_b(x_k,y_l,t^{n+1})+(1-\theta)\,dt\sum_{i=-M_x}^{N_x}\sum_{j=-M_y}^{N_y}u_{ij}^{n}\left[\left(\frac{1}{h_x}\phi_x''(x_k)\delta_{ik}^{(1)}+\frac{1}{h_x^2}\phi_x'(x_k)^2\delta_{ik}^{(2)}\right)\delta_{jl}^{(0)}\right.\\
&\left.+\left(\frac{1}{h_y}\phi_y''(y_l)\delta_{jl}^{(1)}+\frac{1}{h_y^2}\phi_y'(y_l)^2\delta_{jl}^{(2)}\right)\delta_{ik}^{(0)}+V(x_k,y_l)\delta_{ik}^{(0)}\delta_{jl}^{(0)}\right]-i\,\psi_b(x_k,y_l,t^n)\\
&+\theta\,dt\left[(\psi_b)_{xx}(x_k,y_l,t^{n+1})+(\psi_b)_{yy}(x_k,y_l,t^{n+1})+V(x_k,y_l)\psi_b(x_k,y_l,t^{n+1})\right]\\
&+(1-\theta)\,dt\left[(\psi_b)_{xx}(x_k,y_l,t^{n})+(\psi_b)_{yy}(x_k,y_l,t^{n})+V(x_k,y_l)\psi_b(x_k,y_l,t^{n})\right].
\end{aligned}
\tag{2.1.14}
$$

We use the same notations as in [47]: For $z=x$ or y, define $I_{m_z}^{(n)}\equiv\left[\delta_{jk}^{(n)}\right],\ n=0,1,2$, to be the $m_z\times m_z$ Toeplitz matrix, where $m_z=M_z+N_z+1$. Define as well $D(g_z)$ (g_z is a function based on z) to be the $m_z\times m_z$ diagonal matrix, whose diagonal entries are $g_z(z_{-M_z}),\ldots,g_z(z_{N_z})$, where $z_i=\phi_z^{-1}(ih_z)$, $i=-M_z,\ldots,N_z$. In addition, let $U^n,\mathcal{V}$ to be the $m_x\times m_y$ matrices whose $ij-$th entries are u_{ij}^n, $V_{ij}=V(x_i,y_j)$. Besides, assume that J^n, K^n, L^n are $m_x\times m_y$ matrices whose $ij-$th entries are $\psi_b(x_i,y_j,t^n)$, $(\psi_b)_{xx}(x_i,y_j,t^n)$, $(\psi_b)_{yy}(x_i,y_j,t^n)$, respectively. We use the symbol $(.*)$ to show the componentwise multiplication of a constant to the matrix and the symbol $(\circ)$ to represent the Hadamard matrix multiplication.

Following these notations, the system (2.1.14) can be written in the matrix form as follows:

$$
\begin{aligned}
&-i.*U^{n+1}-\theta\,dt.*\left(A^\top U^{n+1}+U^{n+1}B+\mathcal{V}\circ U^{n+1}\right)=i.*J^{n+1}+\theta\,dt.*\left(K^{n+1}+L^{n+1}+\mathcal{V}\circ J^{n+1}\right)\\
&-i.*U^n+(1-\theta)\,dt.*\left(A^\top U^n+U^nB+\mathcal{V}\circ U^n\right)-i.*J^n+(1-\theta)\,dt.*\left(K^n+L^n+\mathcal{V}\circ J^n\right),
\end{aligned}
\tag{2.1.15}
$$

in which the matrices A and B are given by the following expressions:

$$
A=\frac{1}{h_x}.*\left(I_{m_x}^{(1)}\,D\left(\phi_x''\right)\right)+\frac{1}{h_x^2}.*\left(I_{m_x}^{(2)}\,D\left(\left(\phi_x'\right)^2\right)\right),
\tag{2.1.16}
$$

$$
B=\frac{1}{h_y}.*\left(I_{m_y}^{(1)}\,D\left(\phi_y''\right)\right)+\frac{1}{h_y^2}.*\left(I_{m_y}^{(2)}\,D\left(\left(\phi_y'\right)^2\right)\right).
\tag{2.1.17}
$$

In the particular case where the potential function is separable, i.e. $V(x,y)=p(x)+q(y)$, the

system (2.1.15) can be written as:

$$\begin{aligned} -i.*U^{n+1}-\theta\,dt.*\left[\left(A^{\top}+D(p)\right)U^{n+1}+U^{n+1}\left(B+D(q)\right)\right] &= i.*J^{n+1} \\ -i.*U^{n}+(1-\theta)\,dt.*\left[\left(A^{\top}+D(p)\right)U^{n}+U^{n}\left(B+D(q)\right)\right]-i.*J^{n} & \\ +\theta\,dt.*\left(K^{n+1}+L^{n+1}+\mathcal{V}\circ J^{n+1}\right)+(1-\theta)\,dt.*\left(K^{n}+L^{n}+\mathcal{V}\circ J^{n}\right), & \end{aligned} \tag{2.1.18}$$

where $D(p)$ and $D(q)$ are diagonal matrices whose entries are the values of the functions p and q evaluated at the nodes $(x_i)_{i=-M_x}^{N_x}, (y_j)_{j=-M_y}^{N_y}$, respectively.

The matrix system (2.1.18) takes the form of a Sylvester equation:

$$RU^{n+1}+U^{n+1}S=H, \tag{2.1.19}$$

with $R=-i.*I_{m_x}^{(0)}-\theta\,dt.*\left(A^{\top}+D(p)\right)$, $S=-\theta\,dt.*\left(B+D(q)\right)$, and H is the right hand side of (2.1.18). The Sylvester equation (2.1.19) can be solved using the Bartels-Stewart schemes.

One last note that needs to be mentioned is the computation of the unknown matrix U at the first time step. Using the initial condition (2.0.3), the entries of the matrix U can be given by:

$$\left(U^{t^0}\right)_{ij}=f(x_i,y_j)-\psi_b(x_i,y_j,t^0), \tag{2.1.20}$$

with t^0 the first time step.

2.2 Double exponential Sinc-Galerkin method

In order to solve the equation (2.0.1) and determine the unknown coefficients, the Sinc-Galerkin method requires that the residual should be orthogonal to the product of the basis functions, that is:

$$\left(P(\psi_A)-G(\psi_A),S_kS_l\right)=\left(P(\psi_s+\psi_b)-G(\psi_s+\psi_b),S(k,h_x)\circ\phi_x\,S(l,h_y)\circ\phi_y\right)=0, \tag{2.2.1}$$

where

$$P(\psi)=-i\,\psi^{n+1}-\theta\,dt\left((\psi_{xx})^{n+1}+(\psi_{yy})^{n+1}+V(x,y)\psi^{n+1}\right), \tag{2.2.2}$$

$$G(\psi)=-i\,\psi^{n}+(1-\theta)\,dt\left((\psi_{xx})^{n}+(\psi_{yy})^{n}+V(x,y)\psi^{n}\right), \tag{2.2.3}$$

and the inner product $(.,.)$ of two functions f and g on $\Omega=(a,b)\times(a,b)$ is defined by:

$$(f,g)=\int_a^b\int_a^b f(x,y)g(x,y)v(x)w(y)\,dx\,dy, \tag{2.2.4}$$

with $v(x)w(y)$ a weight function. Then, we have:

$$\begin{aligned}(P(\psi_A)-G(\psi_A),S_kS_l) &= -i\int_a^b\int_a^b(\psi_s+\psi_b)^{n+1}S_kS_l\,v(x)w(y)\,dx\,dy\\ &+i\int_a^b\int_a^b(\psi_s+\psi_b)^nS_kS_l\,v(x)w(y)\,dx\,dy-\theta\,dt\int_a^b\int_a^b\Big((\psi_s+\psi_b)_{xx}^{n+1}+(\psi_s+\psi_b)_{yy}^{n+1}\\ &+V(x,y)\,(\psi_s+\psi_b)^{n+1}\Big)S_kS_l\,v(x)w(y)\,dx\,dy-(1-\theta)\,dt\int_a^b\int_a^b\Big(\,(\psi_s+\psi_b)_{xx}^n\\ &+(\psi_s+\psi_b)_{yy}^n+V(x,y)\,(\psi_s+\psi_b)^n\Big)S_kS_l\,v(x)w(y)\,dx\,dy=0.\end{aligned}\tag{2.2.5}$$

Using the integration by parts with v and w chosen such that the boundary terms:

$$b_T^{n+1}=\theta\,dt\int_{\partial\Omega}\left[\psi_s^{n+1}(x,y)\frac{\partial}{\partial N}\Big(S_k(x)S_l(y)v(x)w(y)\Big)-S_k(x)S_l(y)v(x)w(y)\frac{\partial\psi_s^{n+1}}{\partial N}(x,y)\right]d\sigma,\tag{2.2.6}$$

$$b_T^{n}=(1-\theta)\,dt\int_{\partial\Omega}\left[\psi_s^{n}(x,y)\frac{\partial}{\partial N}\Big(S_k(x)S_l(y)v(x)w(y)\Big)-S_k(x)S_l(y)v(x)w(y)\frac{\partial\psi_s^{n}}{\partial N}(x,y)\right]d\sigma\tag{2.2.7}$$

vanish, where N is the outward normal direction to Ω, we obtain:

$$\begin{aligned}&-i\int_a^b\int_a^b(\psi_s+\psi_b)^{n+1}S_kS_lv(x)w(y)\,dx\,dy+i\int_a^b\int_a^b(\psi_s+\psi_b)^nS_kS_lv(x)w(y)\,dx\,dy\\ &-\theta\,dt\int_a^b\int_a^b(\psi_s)^{n+1}\,\Delta\left(S_kS_lv(x)w(y)\right)\,dx\,dy-(1-\theta)\,dt\int_a^b\int_a^b(\psi_s)^n\,\Delta\left(S_kS_lv(x)w(y)\right)\,dx\,dy\\ &-\theta\,dt\int_a^b\int_a^b\left(\left(\,(\psi_b)_{xx}\,\right)^{n+1}+\Big((\psi_b)_{yy}\Big)^{n+1}+V(x,y)\,(\psi_s+\psi_b)^{n+1}\right)S_kS_lv(x)w(y)\,dx\,dy\\ &-(1-\theta)\,dt\int_a^b\int_a^b\left(\left(\,(\psi_b)_{xx}\,\right)^{n}+\Big((\psi_b)_{yy}\Big)^{n}+V(x,y)\,(\psi_s+\psi_b)^{n}\right)S_kS_lv(x)w(y)\,dx\,dy=0.\end{aligned}\tag{2.2.8}$$

Applying the quadrature rule [1] to the inner products, we get:

$$
\begin{aligned}
&- i\,h_x h_y \frac{v(x_k)}{\phi_x'(x_k)} u_{kl}^{n+1} \frac{w(y_l)}{\phi_y'(y_l)} - \theta\, dt\, h_x h_y \left[\frac{w(y_l)}{\phi_y'(y_l)} \sum_{p=-M_x}^{N_x} a_{kp} u_{pl}^{n+1} + \frac{v(x_k)}{\phi_x'(x_k)} \sum_{q=-M_y}^{N_y} b_{lq} u_{kq}^{n+1} \right. \\
&\left. + \frac{v(x_k)}{\phi_x'(x_k)} V(x_k, y_l) u_{kl}^{n+1} \frac{w(y_l)}{\phi_y'(y_l)} \right] = i\,h_x h_y \frac{v(x_k)}{\phi_x'(x_k)} \psi_b(x_k, y_l, t^{n+1}) \frac{w(y_l)}{\phi_y'(y_l)} - i\,h_x h_y \frac{v(x_k)}{\phi_x'(x_k)} u_{kl}^{n} \frac{w(y_l)}{\phi_y'(y_l)} \\
&\quad + \theta\, dt\, h_x h_y \frac{v(x_k)}{\phi_x'(x_k)} \Big[(\psi_b)_{xx}(x_k, y_l, t^{n+1}) + (\psi_b)_{yy}(x_k, y_l, t^{n+1}) + V(x_k, y_l) \psi_b(x_k, y_l, t^{n+1}) \Big] \frac{w(y_l)}{\phi_y'(y_l)} \\
&\quad + (1-\theta)\, dt\, h_x h_y \left[\frac{w(y_l)}{\phi_y'(y_l)} \sum_{p=-M_x}^{N_x} a_{kp} u_{pl}^{n} + \frac{v(x_k)}{\phi_x'(x_k)} \sum_{q=-M_y}^{N_y} b_{lq} u_{kq}^{n} + \frac{v(x_k)}{\phi_x'(x_k)} V(x_k, y_l) u_{kl}^{n} \frac{w(y_l)}{\phi_y'(y_l)} \right] \\
&\quad - i\,h_x h_y \frac{v(x_k)}{\phi_x'(x_k)} \psi_b(x_k, y_l, t^{n}) \frac{w(y_l)}{\phi_y'(y_l)} + (1-\theta)\, dt\, h_x h_y \frac{v(x_k)}{\phi_x'(x_k)} \Big[(\psi_b)_{xx}(x_k, y_l, t^{n}) \\
&\quad + (\psi_b)_{yy}(x_k, y_l, t^{n}) + V(x_k, y_l) \psi_b(x_k, y_l, t^{n}) \Big] \frac{w(y_l)}{\phi_y'(y_l)}, \qquad (2.2.9)
\end{aligned}
$$

where

$$
\begin{aligned}
a_{kp} &= \left(\frac{1}{h_x^2} \delta_{kp}^{(2)} \phi_x'(x_p) v(x_p) + \frac{1}{h_x} \delta_{kp}^{(1)} \left(\frac{\phi_x''(x_p) v(x_p)}{\phi_x'(x_p)} + 2v'(x_p) \right) + \delta_{kp}^{(0)} \frac{v''(x_p)}{\phi_x'(x_p)} \right) \\
b_{lq} &= \left(\frac{1}{h_y^2} \delta_{lq}^{(2)} \phi_y'(y_q) w(y_q) + \frac{1}{h_y} \delta_{lq}^{(1)} \left(\frac{\phi_y''(y_q) w(y_q)}{\phi_y'(y_q)} + 2w'(y_q) \right) + \delta_{lq}^{(0)} \frac{w''(y_q)}{\phi_y'(y_q)} \right).
\end{aligned}
$$

Dividing by $h_x h_y$, it follows:

$$
\begin{aligned}
&- i\, \frac{v(x_k)}{\phi_x'(x_k)} u_{kl}^{n+1} \frac{w(y_l)}{\phi_y'(y_l)} - \theta\, dt \left[\frac{w(y_l)}{\phi_y'(y_l)} \sum_{p=-M_x}^{N_x} a_{kp} u_{pl}^{n+1} + \frac{v(x_k)}{\phi_x'(x_k)} \sum_{q=-M_y}^{N_y} b_{lq} u_{kq}^{n+1} \right. \\
&\left. + \frac{v(x_k)}{\phi_x'(x_k)} V(x_k, y_l) u_{kl}^{n+1} \frac{w(y_l)}{\phi_y'(y_l)} \right] = i\, \frac{v(x_k)}{\phi_x'(x_k)} \psi_b(x_k, y_l, t^{n+1}) \frac{w(y_l)}{\phi_y'(y_l)} - i\, \frac{v(x_k)}{\phi_x'(x_k)} u_{kl}^{n} \frac{w(y_l)}{\phi_y'(y_l)} \\
&\quad - i\, \frac{v(x_k)}{\phi_x'(x_k)} \psi_b(x_k, y_l, t^{n}) \frac{w(y_l)}{\phi_y'(y_l)} + (1-\theta)\, dt \left[\frac{w(y_l)}{\phi_y'(y_l)} \sum_{p=-M_x}^{N_x} a_{kp} u_{pl}^{n} + \frac{v(x_k)}{\phi_x'(x_k)} \sum_{q=-M_y}^{N_y} b_{lq} u_{kq}^{n} \right. \\
&\quad \left. + \frac{v(x_k)}{\phi_x'(x_k)} V(x_k, y_l) u_{kl}^{n} \frac{w(y_l)}{\phi_y'(y_l)} \right] + \theta\, dt\, \frac{v(x_k)}{\phi_x'(x_k)} \Big[(\psi_b)_{xx}(x_k, y_l, t^{n+1}) + (\psi_b)_{yy}(x_k, y_l, t^{n+1}) \\
&\quad + V(x_k, y_l) \psi_b(x_k, y_l, t^{n+1}) \Big] \frac{w(y_l)}{\phi_y'(y_l)} + (1-\theta)\, dt\, \frac{v(x_k)}{\phi_x'(x_k)} \Big[(\psi_b)_{xx}(x_k, y_l, t^{n}) \\
&\quad + (\psi_b)_{yy}(x_k, y_l, t^{n}) + V(x_k, y_l) \psi_b(x_k, y_l, t^{n}) \Big] \frac{w(y_l)}{\phi_y'(y_l)}. \qquad (2.2.10)
\end{aligned}
$$

Premultiplying by the diagonal matrix $D(\phi_x')$, postmultipying by $D(\phi_y')$ and utilizing the same

notations used before, the system (2.2.10) can be represented in the following matrix form:

$$
\begin{aligned}
&- i.*D(v)\,U^{n+1}\,D(w) - \theta\,dt.*\Big[D(\phi_x')\,\mathcal{A}(v)\,U^{n+1}\,D(w) + D(v)\,U^{n+1}\,\mathcal{B}(w)^{\top}D(\phi_y') \\
&\quad + D(v)\,\left(\mathcal{V}\circ U^{n+1}\right)\,D(w)\Big] = -i.*D(v)\,U^{n}\,D(w) + (1-\theta)\,dt.*\Big[D(\phi_x')\,\mathcal{A}(v)\,U^{n}\,D(w) \\
&\quad + D(v)\,U^{n}\,\mathcal{B}(w)^{\top}D(\phi_y') + D(v)\,\left(\mathcal{V}\circ U^{n}\right)\,D(w)\Big] + i.*D(v)\,J^{n+1}\,D(w) \\
&\quad + \theta\,dt.*D(v)\Big[K^{n+1}+L^{n+1}+\mathcal{V}\circ J^{n+1}\Big]\,D(w) - i.*D(v)\,J^{n}\,D(w) \\
&\quad + (1-\theta)\,dt.*D(v)\Big[K^{n}+L^{n}+\mathcal{V}\circ J^{n}\Big]\,D(w),
\end{aligned}
\tag{2.2.11}
$$

where

$$
\mathcal{A}(v) = \frac{1}{h_x^2}.*\left(I_{m_x}^{(2)}D(\phi_x'v)\right) + \frac{1}{h_x}.*\left(I_{m_x}^{(1)}D\left(\frac{\phi_x''v}{\phi_x'}+2v'\right)\right) + I_{m_x}^{(0)}D\left(\frac{v''}{\phi_x'}\right), \tag{2.2.12}
$$

$$
\mathcal{B}(w) = \frac{1}{h_y^2}.*\left(I_{m_y}^{(2)}D(\phi_y'w)\right) + \frac{1}{h_y}.*\left(I_{m_y}^{(1)}D\left(\frac{\phi_y''w}{\phi_y'}+2w'\right)\right) + I_{m_y}^{(0)}D\left(\frac{w''}{\phi_y'}\right). \tag{2.2.13}
$$

In the separable case, the system (2.2.11) can be expressed as:

$$
\begin{aligned}
&- i.*\lambda^{n+1} - \theta\,dt.*\Big[\,(A+D(p))\,\lambda^{n+1} + \lambda^{n+1}\,(B+D(q))\,\Big] = -i.*\lambda^{n} + i.*D(v)\,J^{n+1}\,D(w) \\
&\quad + (1-\theta)\,dt.*\Big[\,(A+D(p))\,\lambda^{n} + \lambda^{n}\,(B+D(q))\,\Big] + \theta\,dt.*D(v)\Big[K^{n+1}+L^{n+1} \\
&\quad + \mathcal{V}\circ J^{n+1}\Big]\,D(w) - i.*D(v)\,J^{n}\,D(w) + (1-\theta)\,dt.*D(v)\Big[K^{n}+L^{n}+\mathcal{V}\circ J^{n}\Big]\,D(w),
\end{aligned}
\tag{2.2.14}
$$

where

$$
\lambda^{n+1} = D(v)\,U^{n+1}\,D(w),\ \lambda^{n} = D(v)\,U^{n}\,D(w),\ A = D(\phi_x')\,\mathcal{A}(v)\,D\left(\frac{1}{v}\right), B = D\left(\frac{1}{w}\right)\,\mathcal{B}(w)^{\top}D(\phi_y'). \tag{2.2.15}
$$

Finally, we can get the matrix U^{n+1} using the following expression:

$$
U^{n+1} = D\left(\frac{1}{v}\right)\lambda^{n+1}D\left(\frac{1}{w}\right), \tag{2.2.16}
$$

where λ^{n+1} is obtained by solving the Sylvester equation (2.2.14).

On the other hand, a choice of the weight functions $v(x)$ and $w(y)$ needs to be made. To obtain the standard Sinc-Galerkin method, we choose:

$$
v(x) = (\phi_x'(x))^{-m},\ w(y) = (\phi_y'(y))^{-m}, \tag{2.2.17}
$$

with $m = 1$, and $m = 1/2$ to obtain the symmetric Sinc-Galerkin method.

2.3 Numerical experiments

In this section, we will examine the efficiency of the Sinc Collocation Method (SCM), Standard Sinc-Galerkin method (StandardSGM) and Symmetric Sinc-Galerkin method (SymmetricSGM) when applied to solve the two-dimensional time-dependent Schrödinger equation. We will compare the performance of the double exponential Sinc methods with the single exponential Sinc methods [47]. We will report the maximum error and the CPU time used for each method and will also show the surface plot of real(Re) and imaginary(Im) parts of exact and approximate solutions of the equation.
For both examples considered below, we choose the positive integer M_x arbitrarily and $M_y = N_y = N_x = M_x$. The study domain is chosen as $\Omega = (0,1) \times (0,1)$, the final time is $T = 1$ and the temporal step length $dt = 0.001$. We select $d = \pi/2$ and $\theta = 1/2$.

2.3.1 Example 1

Consider the Schrödinger equation (2.0.1) with $V(x,y) = 1 - 2/x^2 - 2/y^2$. The exact solution is:

$$\psi(x,y,t) = \exp(it)\, x^2\, y^2, \quad 0 \le x, y \le 1. \tag{2.3.1}$$

We can get the initial and boundary conditions using the exact solution as follow:

$$\psi(x,y,0) = x^2y^2, \;\; \psi(0,y,t) = 0, \;\; \psi(1,y,t) = y^2 \exp(it), \;\; \psi(x,0,t) = 0 \;\text{ and }\; \psi(x,1,t) = x^2 \exp(it). \tag{2.3.2}$$

Using (2.1.6), we get:

$$(\psi - \psi_b)(x,y,t) = \exp(it)\, x\, y\, (x-1)\, (y-1). \tag{2.3.3}$$

Applying the transformations in Table 1.1, we obtain: $\alpha = 1,\ \beta = \pi$ and $\gamma = 1$.

Sinc collocation method

Method	M_x	Max error	CPU time(s)
SESCM	8	$5.72.10^{-4}$	42
	16	$5.47.10^{-5}$	112
	24	$7.73.10^{-6}$	211
	32	$1.91.10^{-6}$	354
DESCM	8	$3.34.10^{-5}$	45
	16	$1.42.10^{-6}$	113
	24	$1.26.10^{-7}$	234
	32	$2.05.10^{-8}$	357

Table 2.1: Numerical results of SESCM and DESCM for different values of M_x (Example 1).

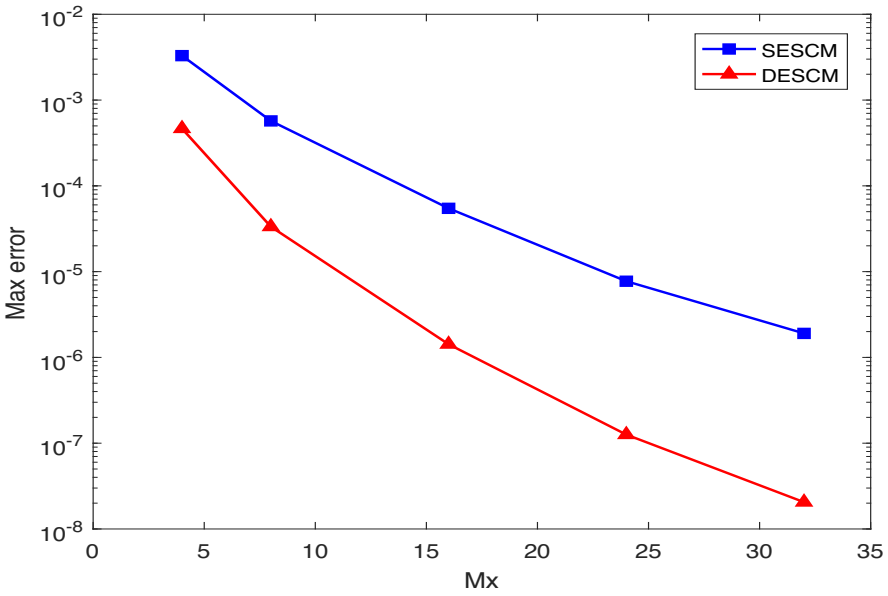

Figure 2.1: Convergence history of SESCM and DESCM (Example 1).

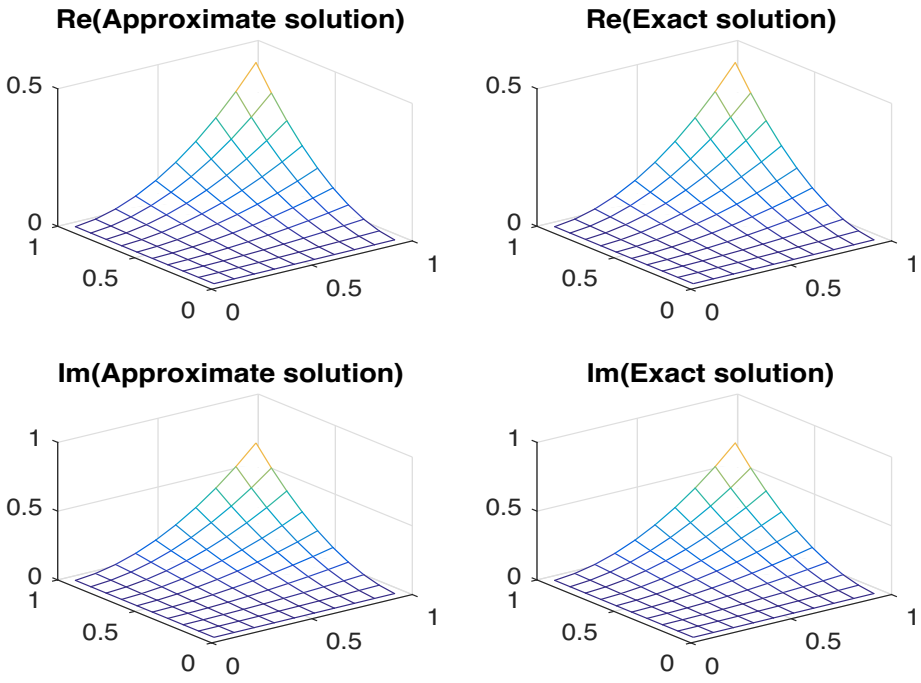

Figure 2.2: Surface plot of real and imaginary parts of exact and approximate solutions for SESCM with $M_x = 8$ (Example 1).

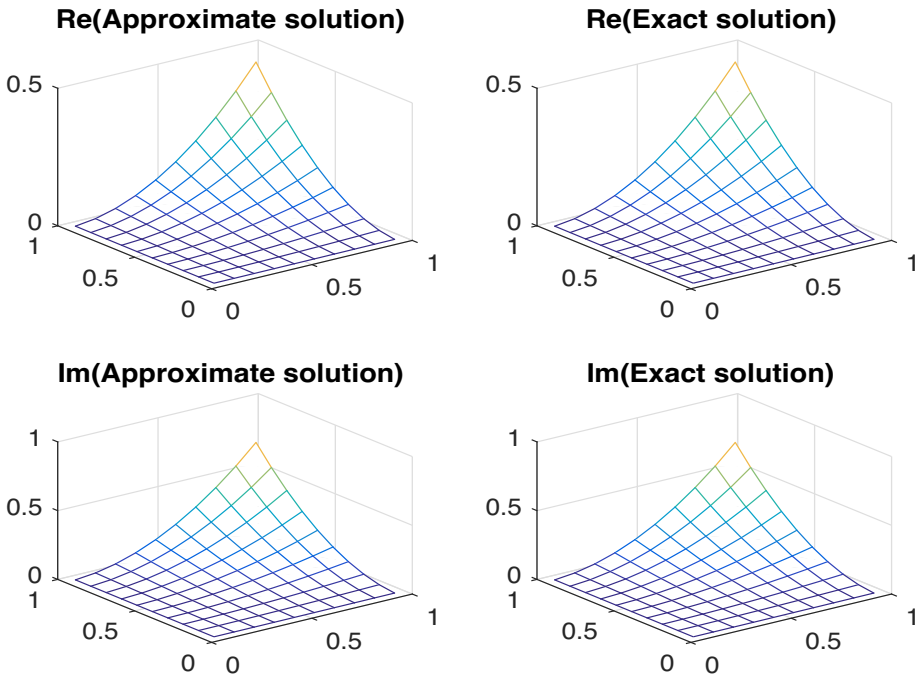

Figure 2.3: Surface plot of real and imaginary parts of exact and approximate solutions for DESCM with $M_x = 8$ (Example 1).

Table 2.1 gives the numerical results for single exponential and double exponential Sinc collocation methods. The table confirms the efficiency of both methods especially that using the double exponential transformation which shows a better performance.

Figure 2.1 curves the maximum error for different values of M_x which gives an idea about the exponential convergence rate of the methods.

Figure 2.2 and Figure 2.3 show the surface plot of real and imaginary parts of exact and approximate solutions of the equation for single exponential Sinc Collocation method(SESCM) and double exponential Sinc Collocation method(DESCM), respectively.

Standard Sinc-Galerkin method

Figure 2.4 curves the maximum error for different values of M_x. Table 2.2 gives the numerical results for single exponential and double exponential standard Sinc-Galerkin methods.

We can clearly deduce from Table 2.2 and Figure 2.4 that the double exponential standard Sinc-Galerkin method (DEStandardSGM) is much more accurate than the single exponential standard Sinc-Galerkin method (SEStandardSGM).

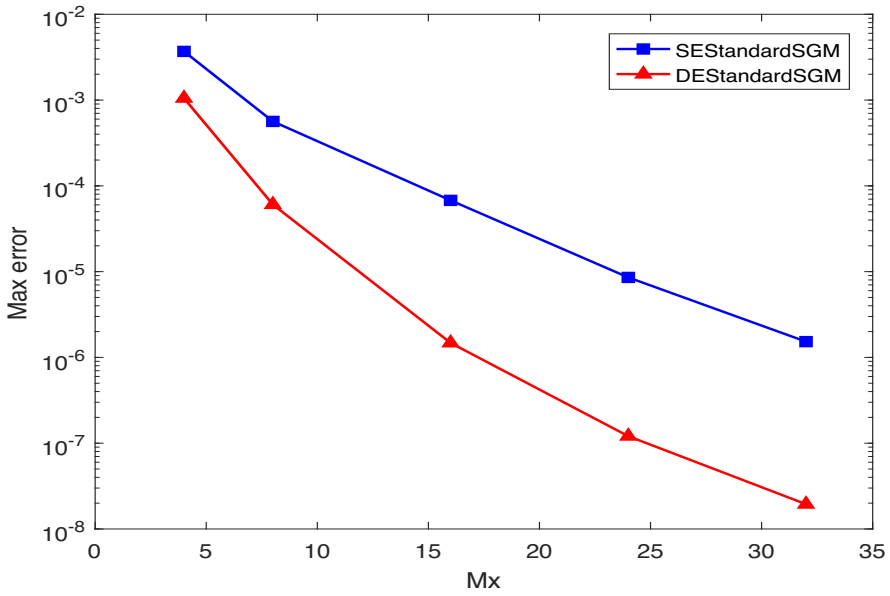

Figure 2.4: Convergence history of SEStandardSGM and DEStandardSGM (Example 1).

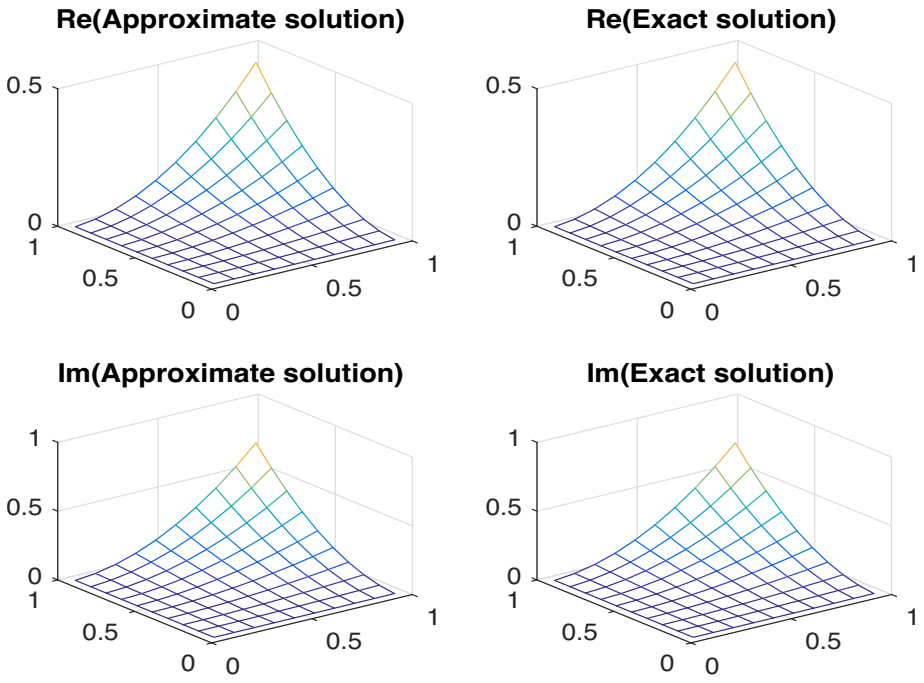

Figure 2.5: Surface plot of real and imaginary parts of exact and approximate solutions for SESTandardSGM with $M_x = 8$ (Example 1).

Method	M_x	Max error	CPU time(s)
SEStandardSGM	8	$5.64.10^{-4}$	48
	16	$6.77.10^{-5}$	117
	24	$8.56.10^{-6}$	231
	32	$1.53.10^{-6}$	366
DEStandardSGM	8	$6.05.10^{-5}$	47
	16	$1.47.10^{-6}$	118
	24	$1.21.10^{-7}$	248
	32	$1.94.10^{-8}$	375

Table 2.2: Numerical results of SEStandardSGM and DEStandardSGM for different values of M_x (Example 1).

Figure 2.5 and Figure 2.6 show the surface plot of real and imaginary parts of exact and approximate solutions of the equation for single exponential Standard Sinc-Galerkin method(SEStandardSGM) and double exponential Standard Sinc-Galerkin method(DEStandardSGM), respectively.

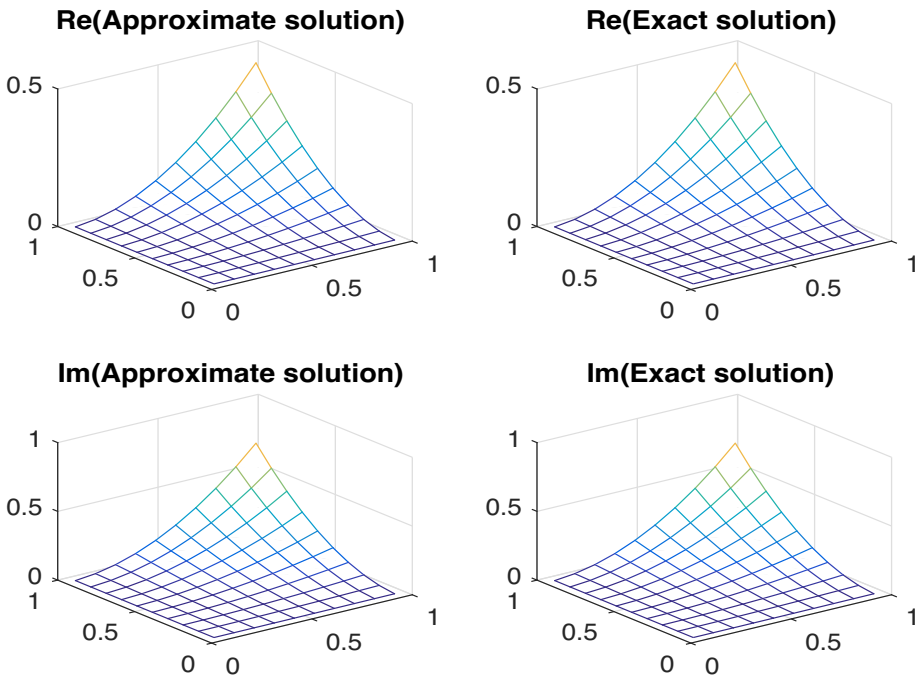

Figure 2.6: Surface plot of real and imaginary parts of exact and approximate solutions for DE-StandardSGM with $M_x = 8$ (Example 1).

Symmetric Sinc-Galerkin method

Figure 2.9 curves the maximum error for different values of M_x. Table 2.3 gives the numerical results for single exponential and double exponential symmetric Sinc-Galerkin methods.

Method	M_x	Max error	CPU time(s)
SESymmetricSGM	8	$6.11.10^{-4}$	40
	16	$5.84.10^{-5}$	107
	24	$9.45.10^{-6}$	219
	32	$1.53.10^{-6}$	370
DESymmetricSGM	8	$4.49.10^{-5}$	45
	16	$1.58.10^{-6}$	115
	24	$1.28.10^{-7}$	229
	32	$1.98.10^{-8}$	360

Table 2.3: Numerical results of SESymmetricSGM and DESymmetricSGM for different values of M_x (Example 1).

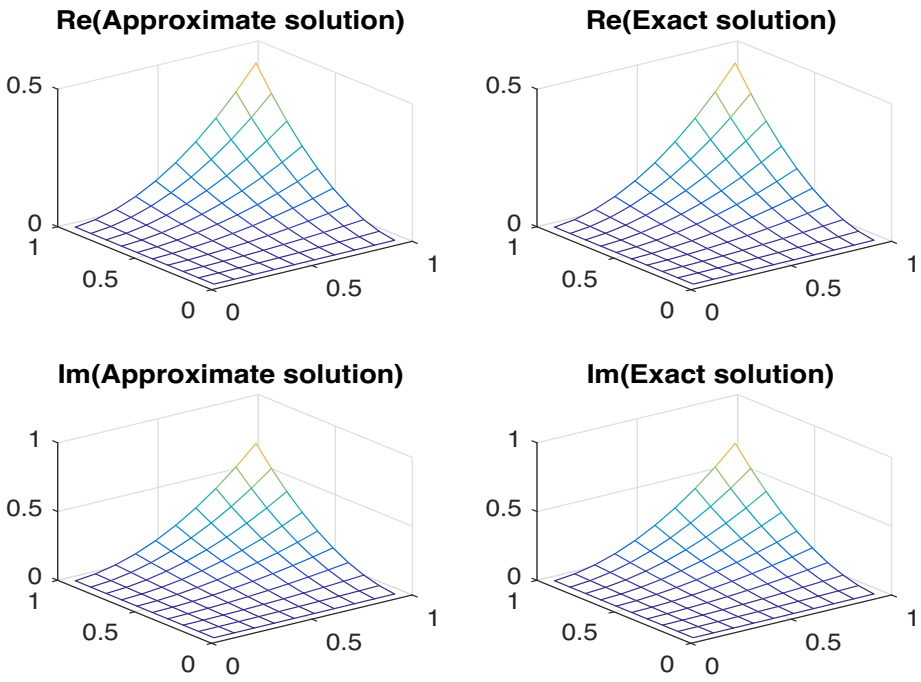

Figure 2.7: Surface plot of real and imaginary parts of exact and approximate solutions for SESymmetricSGM with $M_x = 8$ (Example 1).

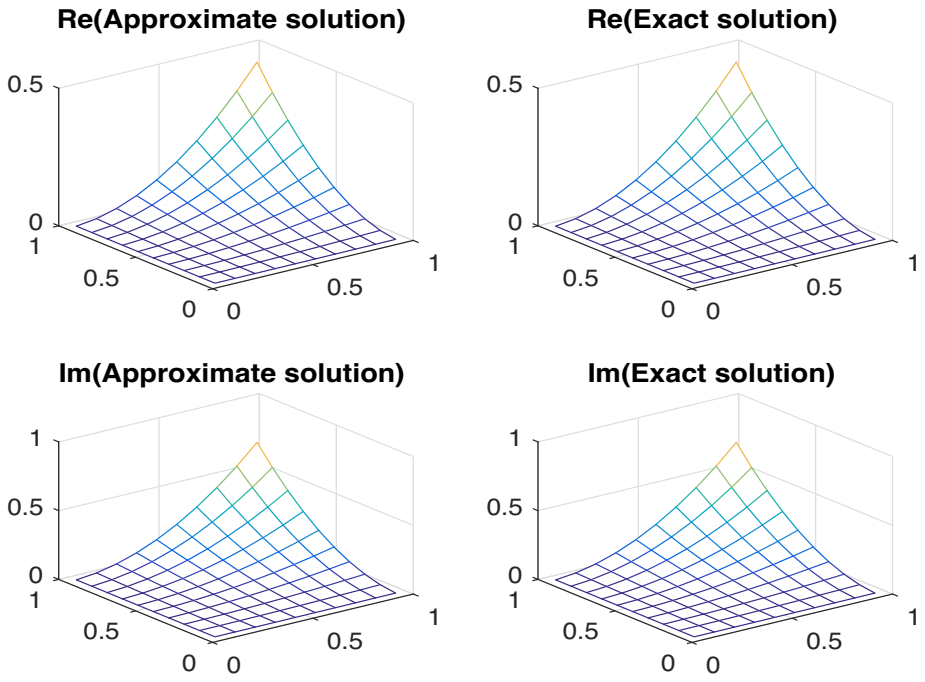

Figure 2.8: Surface plot of real and imaginary parts of exact and approximate solutions for DESymmetricSGM with $M_x = 8$ (Example 1).

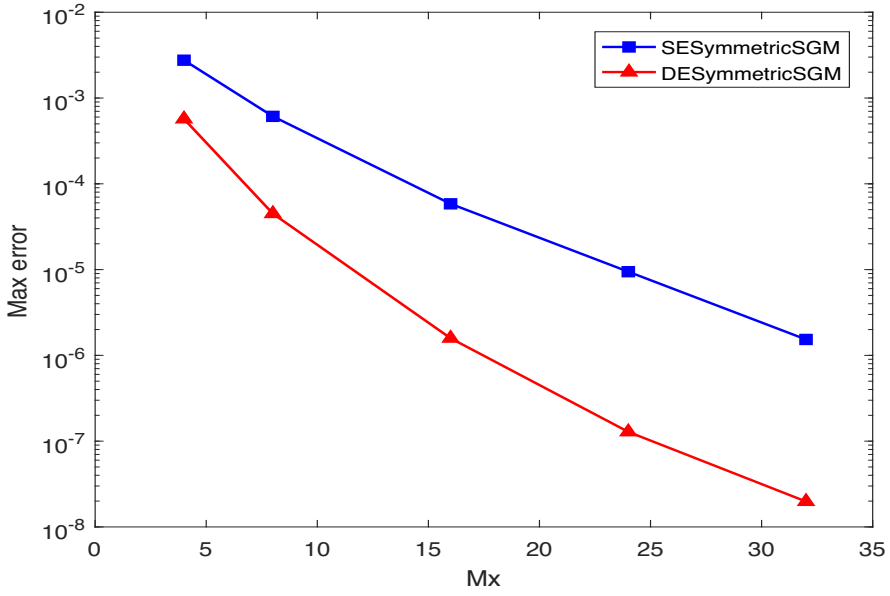

Figure 2.9: Convergence history of SESymmetricSGM and DESymmetricSGM (Example 1).

Figure 2.7 and Figure 2.8 show the surface plot of real and imaginary parts of exact and approximate solutions of the equation for single exponential symmetric Sinc-Galerkin method(SESymmetricSGM) and double exponential symmetric Sinc-Galerkin method(DESymmetricSGM), respectively.

2.3.2 Example 2

Consider the equation (2.0.1) with $V(x,y) = 3 - 2\,tanh^2(x) - 2\,tanh^2(y)$, where the exact solution is:

$$\psi(x,y,t) = \frac{i\,exp(it)}{cosh(x)\,cosh(y)}, \quad 0 \leq x, y \leq 1. \tag{2.3.4}$$

We can get the initial and boundary conditions from the exact solution and we have $\alpha = 1,\ \beta = \pi$ and $\gamma = 1$.

Sinc collocation method

Figure 2.10 curves the maximum error for different values of M_x and Table 2.4 gives the numerical results for single exponential and double exponential Sinc collocation methods.

Figure 2.11 and Figure 2.12 show the surface plot of real and imaginary parts of exact and approximate solutions of the equation for single exponential Sinc Collocation method(SESCM) and double exponential Sinc Collocation method(DESCM), respectively.

Method	M_x	Max error	CPU time(s)
SESCM	8	$1.25.10^{-5}$	55
	16	$1.01.10^{-6}$	167
	24	$1.79.10^{-7}$	311
	32	$3.80.10^{-8}$	514
DESCM	8	$1.29.10^{-6}$	59
	16	$3.02.10^{-8}$	147
	24	$3.3.10^{-9}$	316
	32	$1.42.10^{-9}$	381

Table 2.4: Numerical results of SESCM and DESCM for different values of M_x (Example 2).

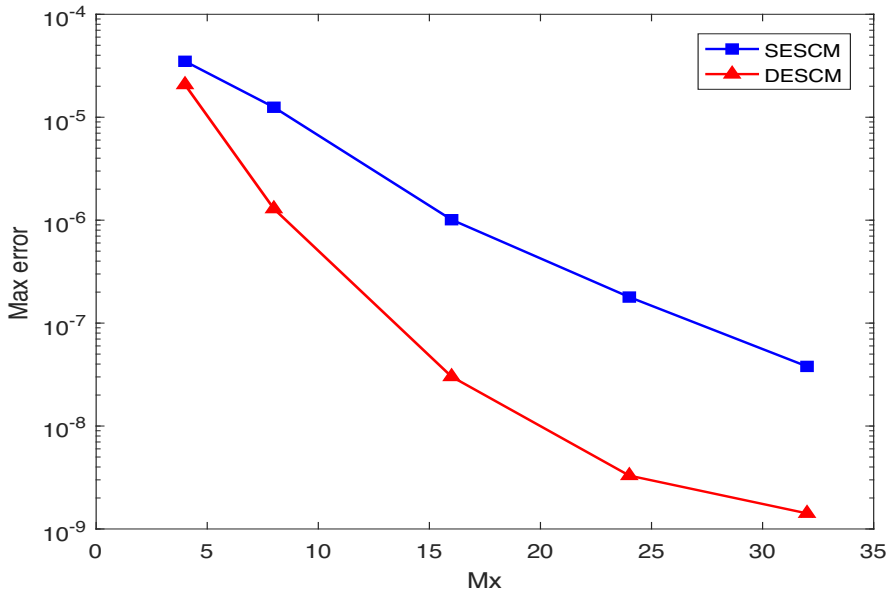

Figure 2.10: Convergence history of SESCM and DESCM (Example 2).

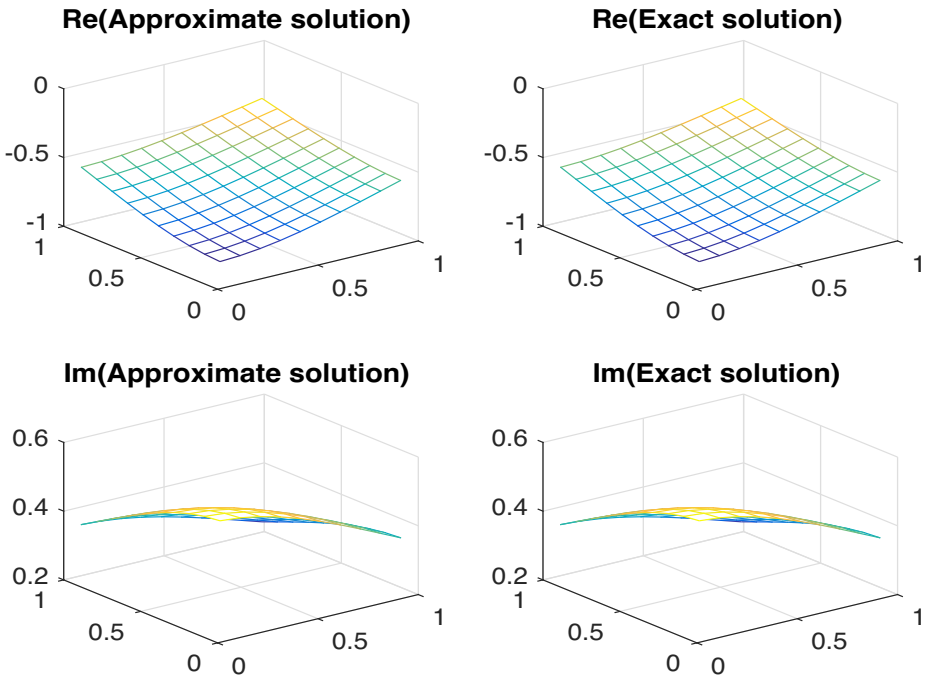

Figure 2.11: Surface plot of real and imaginary parts of exact and approximate solutions for SESCM with $M_x = 16$ (Example 2).

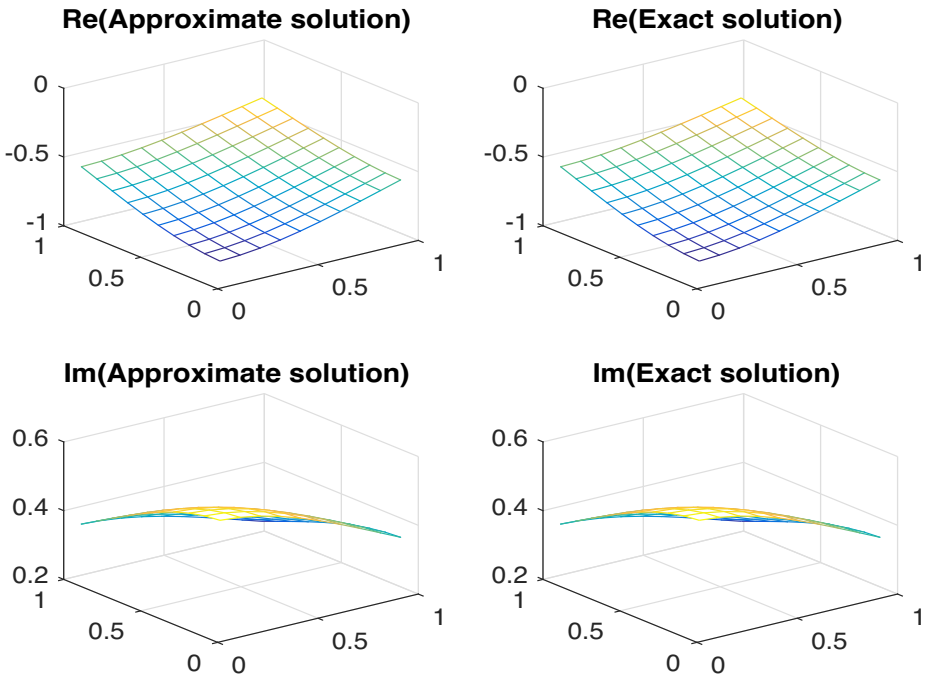

Figure 2.12: Surface plot of real and imaginary parts of exact and approximate solutions for DESCM with $M_x = 16$ (Example 2).

Standard Sinc-Galerkin method

Method	M_x	Max error	CPU time(s)
SEStandardSGM	8	$1.31.10^{-5}$	58
	16	$1.43.10^{-6}$	156
	24	$1.82.10^{-7}$	315
	32	$3.59.10^{-8}$	501
DEStandardSGM	8	$1.43.10^{-6}$	59
	16	$3.49.10^{-8}$	222
	24	$3.55.10^{-9}$	311
	32	$1.39.10^{-9}$	507

Table 2.5: Numerical results of SEStandardSGM and DEStandardSGM for different values of M_x (Example 2).

Table 2.5 shows the numerical results for single exponential and double exponential standard Sinc-Galerkin methods. Figure 2.13 curves the maximum error for different values of M_x for these methods. Figure 2.14 and Figure 2.15 show the surface plot of real and imaginary parts of exact and approximate solutions of the equation for single exponential stan-

dard Sinc-Galerkin method(SEStandardSGM) and double exponential standard Sinc-Galerkin method(DEStandardSGM), respectively.

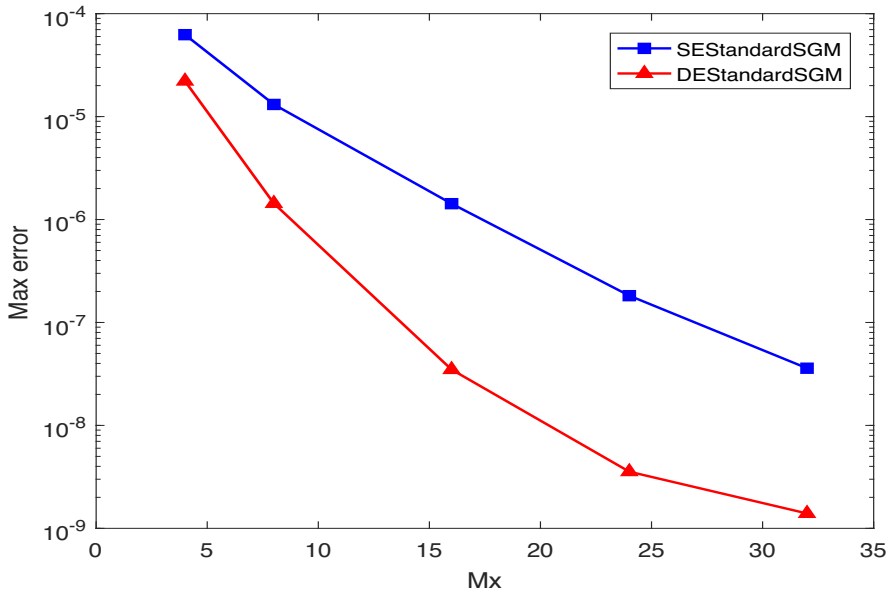

Figure 2.13: Convergence history of SEStandardSGM and DEStandardSGM (Example 2).

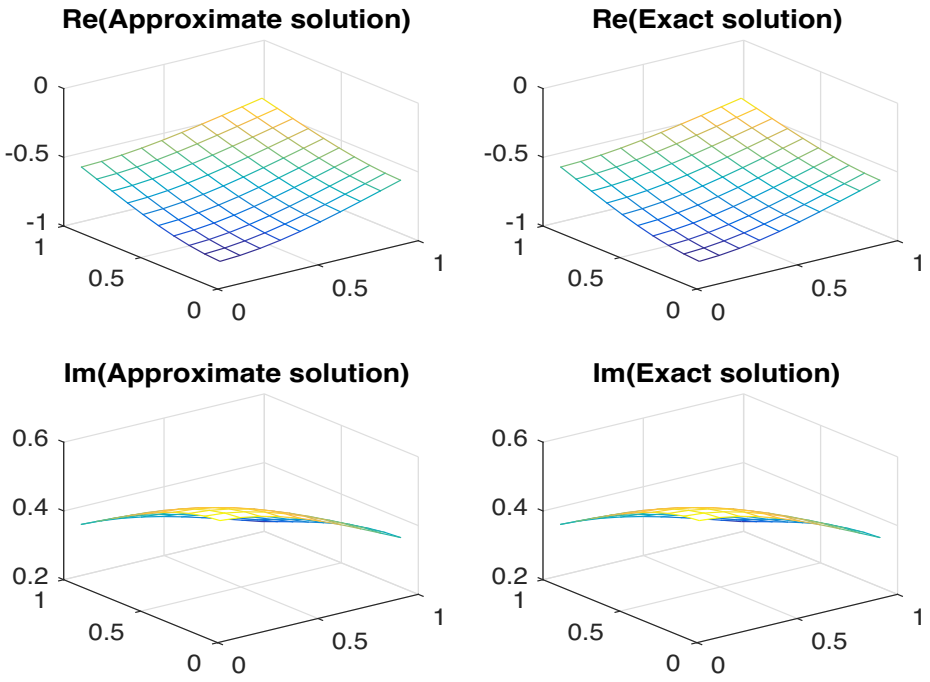

Figure 2.14: Surface plot of real and imaginary parts of exact and approximate solutions for SESTandardSGM with $M_x = 16$ (Example 2).

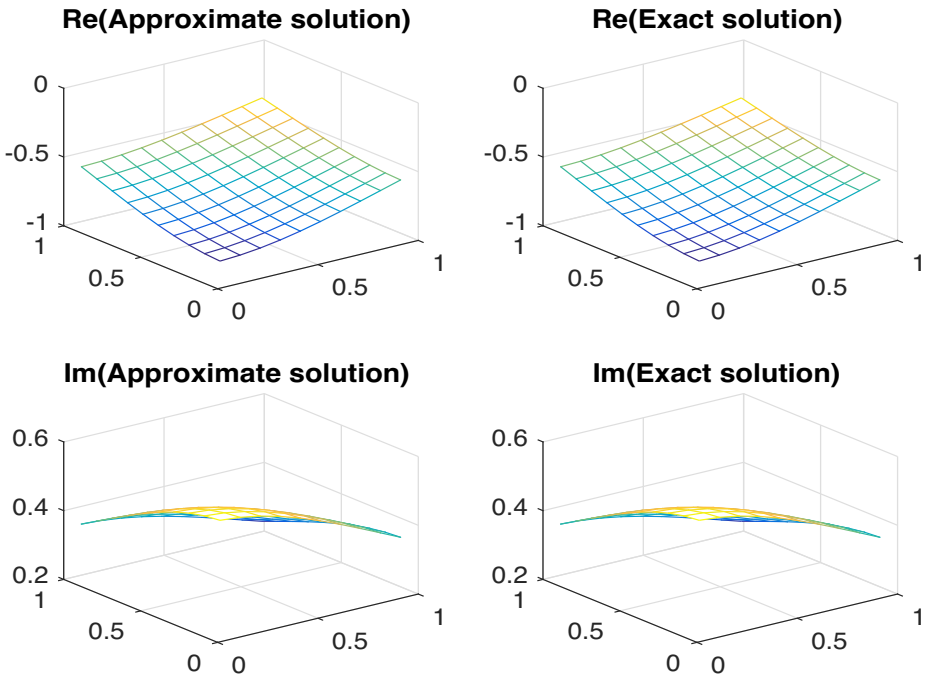

Figure 2.15: Surface plot of real and imaginary parts of exact and approximate solutions for DE-StandardSGM with $M_x = 16$ (Example 2).

Symmetric Sinc-Galerkin method

Method	M_x	Max error	CPU time(s)
SESymmetricSGM	8	$1.19.10^{-5}$	56
	16	$1.19.10^{-6}$	151
	24	$1.86.10^{-7}$	304
	32	$3.48.10^{-8}$	489
DESymmetricSGM	8	$1.41.10^{-6}$	57
	16	$3.49.10^{-8}$	149
	24	$3.74.10^{-9}$	304
	32	$1.35.10^{-9}$	503

Table 2.6: Numerical results of SESymmetricSGM and DESymmetricSGM for different values of M_x (Example 2).

Table 2.6 shows the numerical results for single exponential and double exponential symmetric Sinc-Galerkin methods. Figure 2.16 curves the maximum error for different values of M_x.

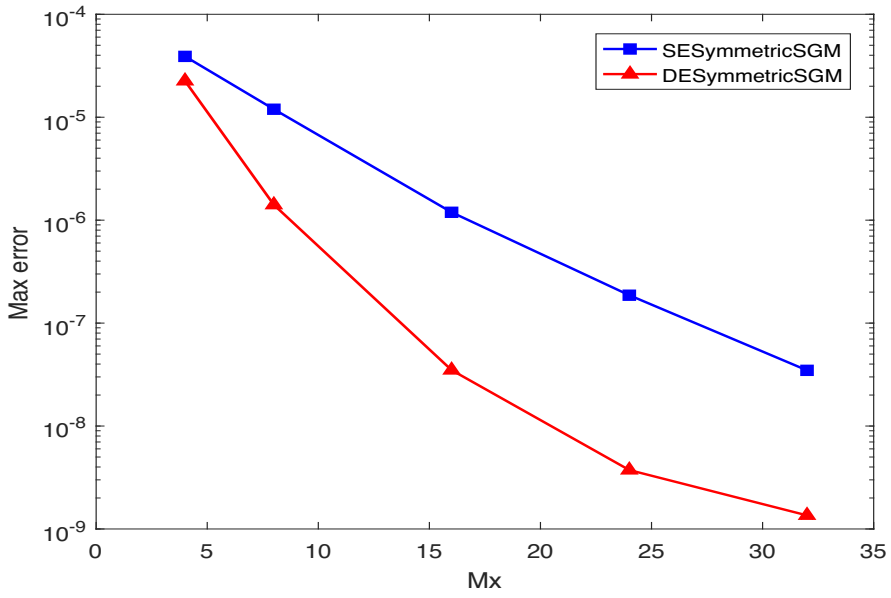

Figure 2.16: Convergence history of SESymmetricSGM and DESymmetricSGM (Example 2).

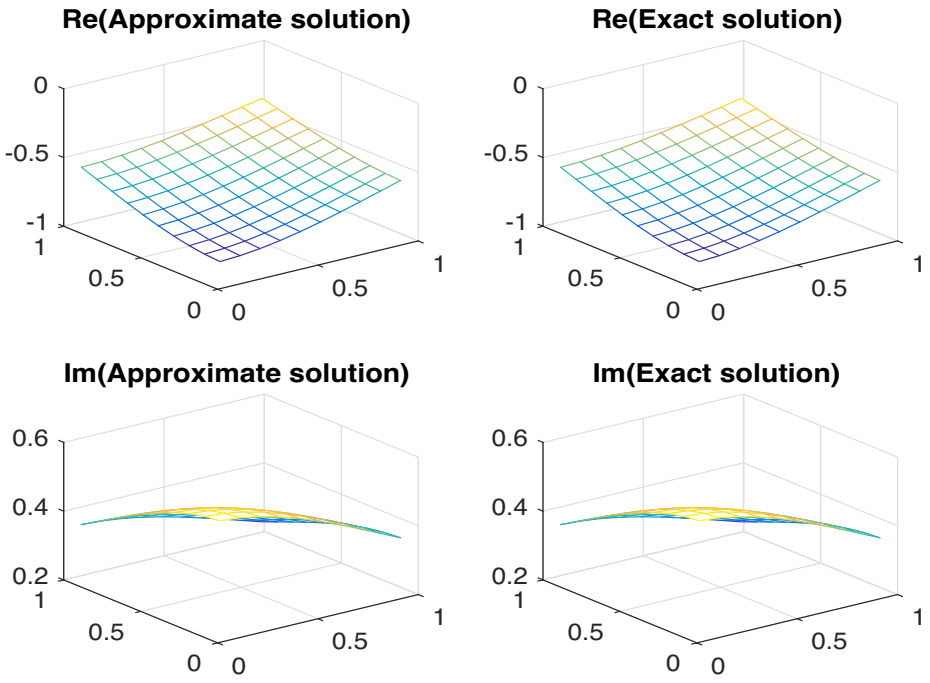

Figure 2.17: Surface plot of real and imaginary parts of exact and approximate solutions for SESymmetricSGM with $M_x = 16$ (Example 2).

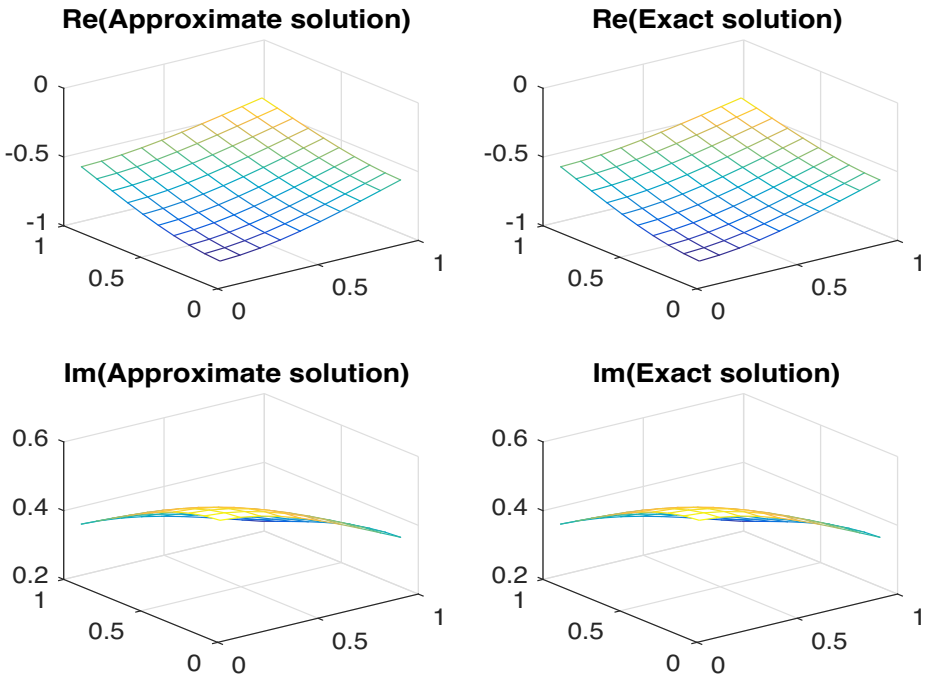

Figure 2.18: Surface plot of real and imaginary parts of exact and approximate solutions for DESymmetricSGM with $M_x = 16$ (Example 2).

Figure 2.17 and Figure 2.18 show the surface plot of real and imaginary parts of exact and approximate solutions of the equation for single exponential symmetric Sinc-Galerkin method(SESymmetricSGM) and double exponential symmetric Sinc-Galerkin method(DESymmetricSGM), respectively.

Conclusion

In this chapter, we applied the Sinc collocation method and the Sinc-Galerkin method incorporated with double exponential transformation in order to solve the time-dependent Schrödinger equation with nonhomogeneous initial and boundary conditions in two dimensions. We compared the proposed approaches with those using single exponential transformation. According to the tables and figures presented above, we can conclude that all methods are reliable and efficient, particularly those using the double exponential transformation which give better results and much more performance accuracy than the methods using single exponential transformation.

Chapter 3

Double exponential Sinc methods for the time-independent Schrödinger equation

The time-independent Schrödinger equation is expressed as:

$$\mathcal{H}\,\psi(\mathbf{x}) = E\,\psi(\mathbf{x}) \qquad \text{for} \qquad \mathbf{x} \in \Omega \tag{3.0.1}$$

$$\psi(\mathbf{x}) = 0 \qquad \text{for} \qquad \mathbf{x} \in \partial\Omega, \tag{3.0.2}$$

where ψ is the unknown wave function, E is the energy eigenvalue, $\partial\Omega$ is the boundary of Ω, and $\mathcal{H}$ stands for the Hamiltonian operator which is made up with the kinetic energy operator Δ and the potential energy operator V:

$$\mathcal{H} = -\Delta + V. \tag{3.0.3}$$

The Schrödinger equation was one of the greatest achievement of twentieth century science. Solving this equation involves finding the energy levels of quantum mechanical systems by finding the energy eigenvalues E of the eigenvalue problem (3.0.1). Obtaining the exact solutions of the Schrodinger equation using analytical methods is possible only for few particular potentials. However, for many other potentials, numerical methods must be employed.

The resolution of the two-dimensional Schrödinger equation has been of great interest in mathematical physics [50, 51, 52]. In [50], the Schrödinger equation with one- and two-dimensional potentials were solved in the framework of the $sl_2(\mathbb{R})$ Lie algebra. Eigenfunctions of the Schrödinger equation for various asymmetric double-well potentials have been determined and the eigenstates are expressed in terms of the orthogonal polynomials. Ixaru [51] introduced a new numerical method for the eigenvalue problem arising from the two-dimensional Schrödinger equation. The method consists in a flexible transformation from a two-dimensional problem into a set of one-dimensional single and coupled channel problems. This set of problems is then solved numerically using highly tuned codes based on CP methods. The energy eigenvalues of coupled oscillators in two dimensions with quartic and sextic couplings have been calculated in [52] where the unbounded domain of the wave function has been truncated and various combination of trigonometric functions are employed as the basis sets in a Rayleigh-Ritz variational method.

The three-dimensional Schrödinger equation is of great importance in different phenomena with many applications in physics and chemistry. It is a partial differential equation involving three coordinates per particle. The mathematical difficulty behind such an equation with different potentials has been surmounted by many exact analytical means and approximate numerical methods. Dong [53] applied the ansatz method for analyzing the Schrödinger equation in D dimensions with three anharmonic potentials: sextic potential, singular even-power potential and singular one-fraction power potential. He applied a suitable ansatz to the wave functions to obtain restrictions on the parameters of the potentials. In [54], Ikhdair and Sever found the polynomial solution of the $3D$ Schrödinger equation for the Pseudoharmonic potential for any arbitrary angular momentum. The exact bound-state energy eigenvalues and the corresponding eigenfunctions are analytically calculated. By a proper transformation, this problem is reduced to a harmonic oscillator potential plus an additional centrifugal potential barrier which can be solved by using the known eigensolutions of anharmonic oscillator potential. In [55], Farizky et al. have solved the Schrödinger equation in three dimensions for Eckart and Manning-Rosen non-central potential by using the asymptotic iteration method. They have obtained the unnormalized wave function and the energy spectrum for a particle influenced by this potential for a certain value of quantum numbers.

Despite the amount of attention the equation (3.0.1) has attracted in literature, it hasn't been treated sufficiently using Cartesian coordinates. The current chapter is devoted to numerically handle the stationary equation and the numerical examples are restricted to the particular case where there is no interaction between particles, i.e. when the potential is considered with variables separated. Examples where the variables are nonseparated are given in the next chapter.

3.1 Two-dimensional Schrödinger equation

In this section, we discuss the resolution of the two-dimensional time-independent Schrödinger equation using double exponential Sinc collocation method (DESCM) and double exponential Sinc-Galerkin method (DESGM). The basis functions and the grid points we use are extensions of those used in the one variable study.

3.1.1 Double exponential Sinc collocation method

The DESCM consists in inserting in equation (3.0.1) the approximate solution given by:

$$\psi_A(x,y) = \sum_{i=-M_x}^{N_x} \sum_{j=-M_y}^{N_y} u_{ij} S_i(x) S_j(y), \tag{3.1.1}$$

where $S_i(x) = S(i,h_x) \circ \phi_x(x)$ and $S_j(y) = S(j,h_y) \circ \phi_y(y)$, and calculating the resulting system at the grid points:

$$(x_k, y_l) = \left(\phi_x^{-1}(kh_x), \phi_y^{-1}(lh_y)\right), \tag{3.1.2}$$

for $k = -M_x, \ldots, N_x$ and $l = -M_y, \ldots, N_y$, with ϕ_x and ϕ_y are the double exponential transformations. This results in:

$$-\left(\frac{\partial^2 \psi_A}{\partial x^2} + \frac{\partial^2 \psi_A}{\partial y^2}\right)(x_k, y_l) + V(x_k, y_l)\, \psi_A(x_k, y_l) = \epsilon\, \psi_A(x_k, y_l), \tag{3.1.3}$$

where ϵ is the approximation of the energy eigenvalue E in (3.0.1). Using (3.1.1), we obtain:

$$\sum_{i=-M_x}^{N_x}\sum_{j=-M_y}^{N_y} u_{ij}\left[-S_i''(x_k)S_j(y_l) - S_i(x_k)S_j''(y_l) + V(x_k,y_l)S_i(x_k)S_j(y_l)\right] = \epsilon \sum_{i=-M_x}^{N_x}\sum_{j=-M_y}^{N_y} u_{ij}S_i(x_k)S_j(y_l), \quad (3.1.4)$$

where u_{ij} are the coefficients to be determined. From this, it follows that:

$$\begin{aligned} -\sum_{i=-M_x}^{N_x}\sum_{j=-M_y}^{N_y} u_{ij}\left(\frac{1}{h_x}\phi_x''(x_k)\delta_{ik}^{(1)} + \frac{1}{h_x^2}\phi_x'(x_k)^2\delta_{ik}^{(2)}\right)\delta_{jl}^{(0)} \\ -\sum_{i=-M_x}^{N_x}\sum_{j=-M_y}^{N_y} u_{ij}\delta_{ik}^{(0)}\left(\frac{1}{h_y}\phi_y''(y_l)\delta_{jl}^{(1)} + \frac{1}{h_y^2}\phi_y'(y_l)^2\delta_{jl}^{(2)}\right) \\ +\sum_{i=-M_x}^{N_x}\sum_{j=-M_y}^{N_y} u_{ij}V(x_k,y_l)\delta_{ik}^{(0)}\delta_{jl}^{(0)} = \epsilon\sum_{i=-M_x}^{N_x}\sum_{j=-M_y}^{N_y} u_{ij}\delta_{ik}^{(0)}\delta_{jl}^{(0)}. \end{aligned} \quad (3.1.5)$$

In a matrix form, the above system can be written as:

$$-\left(A^\top U + UB\right) + \mathcal{V}\circ U = \epsilon\, U, \quad (3.1.6)$$

where $(\circ)$ represents the Hadamard matrix multiplication and the matrices A and B are given by:

$$A = \frac{1}{h_x}\cdot_* \left(I_{m_x}^{(1)} D\left(\phi_x''\right)\right) + \frac{1}{h_x^2}\cdot_* \left(I_{m_x}^{(2)} D\left((\phi_x')^2\right)\right), \quad (3.1.7)$$

$$B = \frac{1}{h_y}\cdot_* \left(I_{m_y}^{(1)} D\left(\phi_y''\right)\right) + \frac{1}{h_y^2}\cdot_* \left(I_{m_y}^{(2)} D\left((\phi_y')^2\right)\right), \quad (3.1.8)$$

where ".$*$" denotes the componentwise multiplication of a constant by a matrix, $m_x = M_x + N_x + 1$ and $m_y = M_y + N_y + 1$, U and $\mathcal{V}$ are the $m_x \times m_y$ matrices whose $ij-$th entries are u_{ij} and $V_{ij} = V(x_i, y_j)$, respectively. Furthermore, $I_{m_x}^{(n)}$ and $I_{m_y}^{(n)}$ are the Toeplitz matrices whose elements are $\delta_{ik}^{(n)}$, $\delta_{jl}^{(n)}$ for $i,k = -M_x,\ldots,N_x$, $j,l = -M_y,\ldots,N_y$ and $n = 0,1,2$. Let g_x be a function of x and g_y be a function of y. $D(g_x)$ and $D(g_y)$ are the diagonal matrices whose diagonal entries are $g_x(x_{-M_x}),\ldots,g_x(x_{N_x})$ and $g_y(y_{-M_y}),\ldots,g_y(y_{N_y})$, respectively.

Using the matrix vectorization operator (vec) and the Kronecker product notation $(\otimes)$, the matrix system (3.1.6) can be written as:

$$\left[-\left(I_{m_y}^{(0)}\otimes A^\top + B^\top \otimes I_{m_x}^{(0)}\right) + \mathcal{D}(\mathcal{V})\right]\mathrm{vec}(U) = \epsilon\,\mathrm{vec}(U), \quad (3.1.9)$$

where $\mathcal{D}(\mathcal{V})$ is the $m_x m_y \times m_x m_y$ matrix given by $\mathcal{D}(\mathcal{V}) = diag\left(D(V_{i,-M_y}),\ldots,D(V_{i,N_y})\right)$ and $\mathrm{vec}(U)$ is the $m_x m_y \times 1$ vector defined by $\mathrm{vec}(U) = \left[[u_{i,-M_y}]^\top,\ldots,[u_{i,N_y}]^\top\right]^\top$ for $i = -M_x,\ldots,N_x$.

In the special case where the potential function is separable, i.e. $V(x,y) = V_x(x) + V_y(y)$, the system (3.1.9) takes the following form:

$$\Big[I_{m_y}^{(0)} \otimes \Big(-A^\top + D(V_x)\Big) + \Big(-B^\top + D(V_y)\Big) \otimes I_{m_x}^{(0)}\Big] \operatorname{vec}(U) = \epsilon \operatorname{vec}(U), \tag{3.1.10}$$

which can be easily solved by solving the two smaller eigenvalue problems:

$$\Big(-A^\top + D(V_x)\Big) w_1 = \epsilon_1 w_1 \quad \text{and} \quad \Big(-B^\top + D(V_y)\Big) w_2 = \epsilon_2 w_2, \tag{3.1.11}$$

where $\epsilon = \epsilon_1 + \epsilon_2$ is the approximation of the exact energy eigenvalue E and $\operatorname{vec}(U) = w_2 \otimes w_1$ is the associated eigenfunction.

3.1.2 Double exponential Sinc-Galerkin method

Unless otherwise stated, the same notations given previously are also used in this section.

The inner product of two functions f and g is given by:

$$(f,g) = \iint_\Omega f(x,y) g(x,y) v(x) w(y)\, dx\, dy, \tag{3.1.12}$$

where the product $v(x)w(y)$ serves as a weight function. Using the DESGM, we require the residual:

$$R(x,y) = -\Delta\psi(x,y) + (V(x,y) - E)\,\psi(x,y), \tag{3.1.13}$$

to be orthogonal to each $S_k(x)S_l(y)$, for $-M_x \le k \le N_x$ and $-M_y \le l \le N_y$. That is:

$$(R, S_k S_l) = 0. \tag{3.1.14}$$

More specifically:

$$\begin{aligned}\iint_\Omega \left(-\Delta\psi(x,y) + V(x,y)\psi(x,y)\right) S_k(x) S_l(y) v(x) w(y)\, dx\, dy \\ - E \iint_\Omega \psi(x,y) S_k(x) S_l(y) v(x) w(y)\, dx\, dy = 0.\end{aligned} \tag{3.1.15}$$

In order to remove the derivatives from U, we apply Green's identity:

$$\begin{aligned}-\iint_\Omega \psi(x,y) \Delta(S_k(x)S_l(y)v(x)w(y))\, dx\, dy + b_T + \iint_\Omega V(x,y)\psi(x,y) S_k(x) S_l(y) v(x) w(y)\, dx\, dy \\ = E \iint_\Omega \psi(x,y) S_k(x) S_l(y) v(x) w(y)\, dx\, dy,\end{aligned} \tag{3.1.16}$$

where the boundary term b_T is defined by:

$$b_T = \int_{\partial\Omega} \Big[\psi(x,y) \frac{\partial}{\partial n} \left(S_k(x)S_l(y)v(x)w(y)\right) - S_k(x)S_l(y)v(x)w(y) \frac{\partial\psi}{\partial n}(x,y)\Big] d\sigma, \tag{3.1.17}$$

and n is the outward normal direction to Ω. Assuming that $b_T = 0$, we get:

$$-\iint_\Omega \psi(x,y)\left[(S_k(x)v(x))''\,S_l(y)w(y) + S_k(x)v(x)\,(S_l(y)w(y))''\right]dx\,dy + \iint_\Omega V(x,y)\psi(x,y)S_k(x)S_l(y)v(x)w(y)\,dx\,dy = E\iint_\Omega \psi(x,y)S_k(x)S_l(y)v(x)w(y)\,dx\,dy. \tag{3.1.18}$$

By applying the quadrature rule in [1] and eliminating the error terms, we obtain:

$$\begin{aligned}-h_xh_y\frac{w(y_l)}{\phi_y'(y_l)}\sum_{p=-M_x}^{N_x}\psi(x_p,y_l)\left[\frac{1}{h_x^2}\delta_{kp}^{(2)}\phi_x'(x_p)v(x_p)+\frac{1}{h_x}\delta_{kp}^{(1)}\left(\frac{\phi_x''(x_p)v(x_p)}{\phi_x'(x_p)}+2v'(x_p)\right)+\delta_{kp}^{(0)}\frac{v''(x_p)}{\phi_x'(x_p)}\right]\\ -h_xh_y\frac{v(x_k)}{\phi_x'(x_k)}\sum_{q=-M_y}^{N_y}\psi(x_k,y_q)\left[\frac{1}{h_y^2}\delta_{lq}^{(2)}\phi_y'(y_q)w(y_q)+\frac{1}{h_y}\delta_{lq}^{(1)}\left(\frac{\phi_y''(y_q)w(y_q)}{\phi_y'(y_q)}+2w'(y_q)\right)\right.\\ \left.+\delta_{lq}^{(0)}\frac{w''(y_q)}{\phi_y'(y_q)}\right]+h_xh_y\frac{v(x_k)}{\phi_x'(x_k)}V(x_k,y_l)\psi(x_k,y_l)\frac{w(y_l)}{\phi_y'(y_l)}=h_xh_y.E\,\frac{v(x_k)}{\phi_x'(x_k)}\psi(x_k,y_l)\frac{w(y_l)}{\phi_y'(y_l)}.\end{aligned} \tag{3.1.19}$$

Dividing by h_xh_y and replacing $\psi(x_k,y_l)$ and E by ψ_{kl} and ϵ, respectively, yields:

$$\begin{aligned}-\frac{w(y_l)}{\phi_y'(y_l)}\sum_{p=-M_x}^{N_x}\psi_{pl}\left[\frac{1}{h_x^2}\delta_{kp}^{(2)}\phi_x'(x_p)v(x_p)+\frac{1}{h_x}\delta_{kp}^{(1)}\left(\frac{\phi_x''(x_p)v(x_p)}{\phi_x'(x_p)}+2v'(x_p)\right)+\delta_{kp}^{(0)}\frac{v''(x_p)}{\phi_x'(x_p)}\right]\\ -\frac{v(x_k)}{\phi_x'(x_k)}\sum_{q=-M_y}^{N_y}\psi_{kq}\left[\frac{1}{h_y^2}\delta_{lq}^{(2)}\phi_y'(y_q)w(y_q)+\frac{1}{h_y}\delta_{lq}^{(1)}\left(\frac{\phi_y''(y_q)w(y_q)}{\phi_y'(y_q)}+2w'(y_q)\right)+\delta_{lq}^{(0)}\frac{w''(y_q)}{\phi_y'(y_q)}\right]\\ +\frac{v(x_k)}{\phi_x'(x_k)}V(x_k,y_l)\psi_{kl}\frac{w(y_l)}{\phi_y'(y_l)}=\epsilon\,\frac{v(x_k)}{\phi_x'(x_k)}\psi_{kl}\frac{w(y_l)}{\phi_y'(y_l)}.\end{aligned} \tag{3.1.20}$$

Premultiplying by the diagonal matrix $D(\phi_x')$ and postmultipying by $D(\phi_y')$, the system can be represented in the following matrix form:

$$-\left(D(\phi_x')\mathcal{A}(v)\,\Psi\,D(w)+D(v)\,\Psi\,\mathcal{B}(w)^\top D(\phi_y')\right)+D(v)\,(\mathcal{V}\circ\Psi)\,D(w)=\epsilon\,D(v)\,\Psi\,D(w), \tag{3.1.21}$$

where Ψ is the $m_x \times m_y$ matrix containing the unknowns ψ_{ij} and:

$$\mathcal{A}(v)=\frac{1}{h_x^2}.*\left(I_{m_x}^{(2)}D(\phi_x'v)\right)+\frac{1}{h_x}.*\left(I_{m_x}^{(1)}D\left(\frac{\phi_x''v}{\phi_x'}+2v'\right)\right)+I_{m_x}^{(0)}D\left(\frac{v''}{\phi_x'}\right) \tag{3.1.22}$$

$$\mathcal{B}(w)=\frac{1}{h_y^2}.*\left(I_{m_y}^{(2)}D(\phi_y'w)\right)+\frac{1}{h_y}.*\left(I_{m_y}^{(1)}D\left(\frac{\phi_y''w}{\phi_y'}+2w'\right)\right)+I_{m_y}^{(0)}D\left(\frac{w''}{\phi_y'}\right). \tag{3.1.23}$$

Using the Kronecker product notation, the system (3.1.21) may be expressed as the large sparse system:

$$\left[-\left(I_{m_y}^{(0)}\otimes A+B^\top\otimes I_{m_x}^{(0)}\right)+\mathcal{D}(\mathcal{V})\right]\mathrm{vec}(U)=\epsilon\,\mathrm{vec}(U), \tag{3.1.24}$$

with:

$$U = D(v)\,\Psi\,D(w), \quad A = D(\phi'_x)\,\mathcal{A}(v)\,D\left(\frac{1}{v}\right) \quad \text{and} \quad B = D\left(\frac{1}{w}\right)\,\mathcal{B}(w)^\top\,D(\phi'_y). \tag{3.1.25}$$

In the present work, our interest is in the computation of the energy eigenvalues but if we want to solve the equation for the wave function, Ψ can be recovered from U by:

$$\Psi = D\left(\frac{1}{v}\right)\,U\,D\left(\frac{1}{w}\right). \tag{3.1.26}$$

where U is obtained by solving the discrete system (3.1.24). Finally, in order to eliminate the boundary term b_T, we let the weight functions:

$$v(x) = (\phi'_x(x))^{-1} \quad \text{and} \quad w(y) = (\phi'_y(y))^{-1}. \tag{3.1.27}$$

3.1.3 Numerical results

In this section, we will examine the efficacy of DESCM and DESGM when applied to solve the two-dimensional time-independent Schrödinger equation. We also compare their performance with the corresponding Sinc methods combined with the single exponential transformation, namely SESCM and SESGM. The figures presented in the examples show the absolute errors corresponding to each method in terms of the number of its collocation points. The absolute error is defined by:

$$\text{Absolute error}(N_x) \;=\; |E \,-\, \epsilon(N_x)|, \tag{3.1.28}$$

where E is the exact value of the energy and $\epsilon(N_x)$ is the approximation of the energy eigenvalue, which corresponds to N_x.

The mesh sizes $h_y = h_x = h$ are defined by:

$$h = \frac{W(\pi d\gamma n/\beta)}{\gamma n}, \tag{3.1.29}$$

where W is the Lambert function and n, β and γ are given in [6, Lemma 3.1]. Unless otherwise stated, we choose the positive integer N_x arbitrarily and $N_y = M_y = M_x = N_x$.

Anharmonic oscillator potentials

We present numerical results for the energy values corresponding to the anharmonic oscillator potentials. The domain is chosen as $\Omega = (-\infty, \infty) \times (-\infty, \infty)$. We use the conformal mapping $\sinh(t)$ as a double exponential transformation.

Some of the potentials presented in one-dimensional space in [32], which have known analytic solutions for energy levels are extended to the two-dimensional space as follows:

$$V_1(x, y) = x^2 - 4x^4 + x^6 + y^2 - 4y^4 + y^6 \quad \Rightarrow \quad E_0 = -2 - 2 = -4, \tag{3.1.30}$$

$$V_2(x, y) = 4x^2 - 6x^4 + x^6 + 4y^2 - 6y^4 + y^6 \quad \Rightarrow \quad E_1 = -9 - 9 = -18. \tag{3.1.31}$$

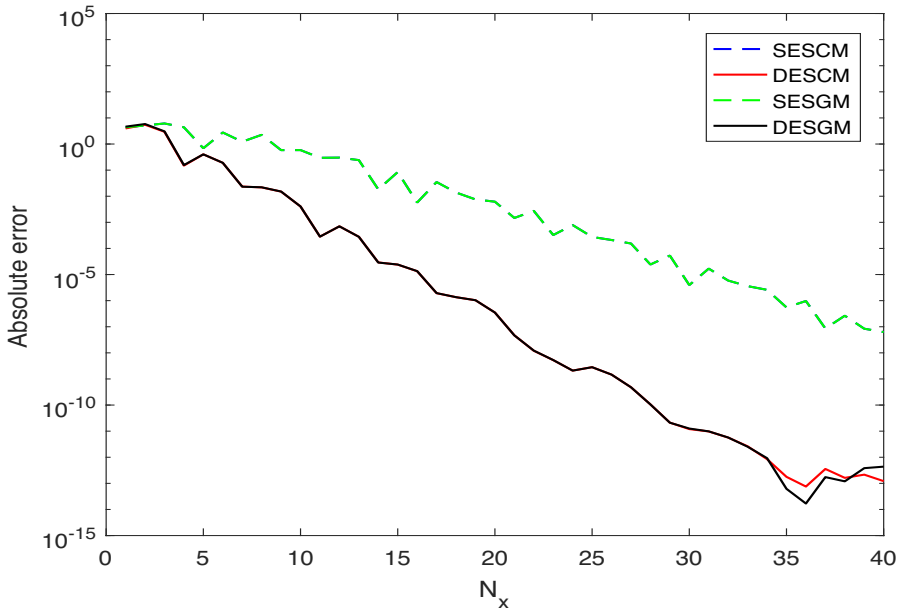

Figure 3.1: Computation of the anharmonic oscillator potential $V_1(x, y)$. The absolute error for the energy ground level E_0 corresponding to $V_1(x, y)$.

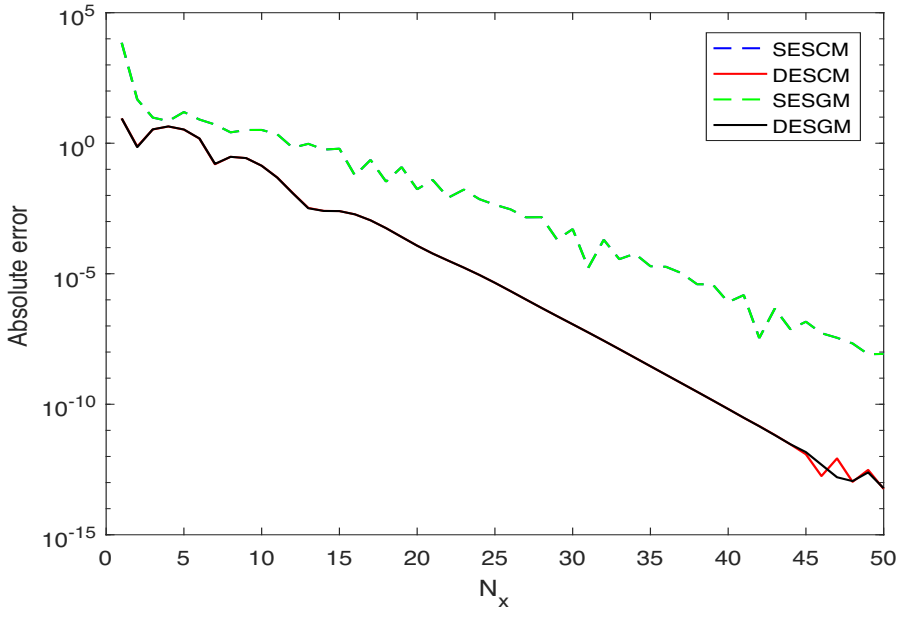

Figure 3.2: Computation of the anharmonic oscillator potential $V_2(x, y)$. The absolute error for the first excited energy E_1 corresponding to $V_2(x, y)$.

In the above calculation, the performance of the SESCM and SESGM is exactly the same.

Anharmonic Coulombic potentials

Consider the two-dimensional Schrödinger equation with anharmonic Coulombic potentials [33]:

$$V_3(x,y) = \frac{2}{x^2} + x^2 + \frac{2}{y^2} + y^2 \Rightarrow E_0 = 5 + 5 = 10, \tag{3.1.32}$$

$$V_4(x,y) = \frac{3}{4x^2} + x^2 + \frac{3}{4y^2} + y^2 \Rightarrow E_0 = 4 + 4 = 8. \tag{3.1.33}$$

We select $\Omega = (0,\infty)\times(0,\infty)$ as our domain and apply the following conformal mapping $\log(e^{\sinh(t)}+1)$.

For the implementation of the DESCM and the DESGM, we choose:

$$M_x = max\left(\left\lfloor \frac{\gamma_R}{\gamma_L} N_x + \frac{log(\beta_R/\beta_L)}{\gamma_L h_x} \right\rfloor, 0\right), \tag{3.1.34}$$

for an arbitrary N_x, where $\beta_L = 1$, $\beta_R = \frac{1}{8}$, $\gamma_L = 1$, $\gamma_R = 2$ and $N_y = N_x$, $M_y = M_x$.

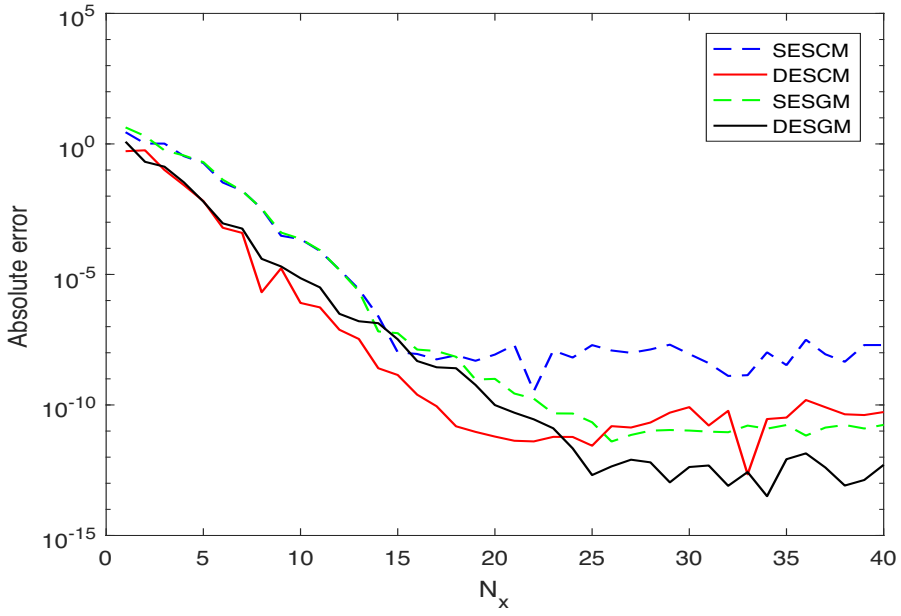

Figure 3.3: Computation of the anharmonic Coulombic potential $V_3(x,y)$. The absolute error for the energy ground level E_0 corresponding to $V_3(x,y)$.

Perturbed anharmonic Coulombic potentials

Consider the anharmonic Coulombic potentials with singular perturbation [34] of the form:

$$V_5(x,y) = x^2 + \frac{9}{64x^6} + y^2 + \frac{9}{64y^6} \;\Rightarrow\; E_0 = 4+4 = 8. \tag{3.1.35}$$

We select the domain Ω as $(0,\infty)\times(0,\infty)$ and use the following conformal mapping: $\exp(t)$.

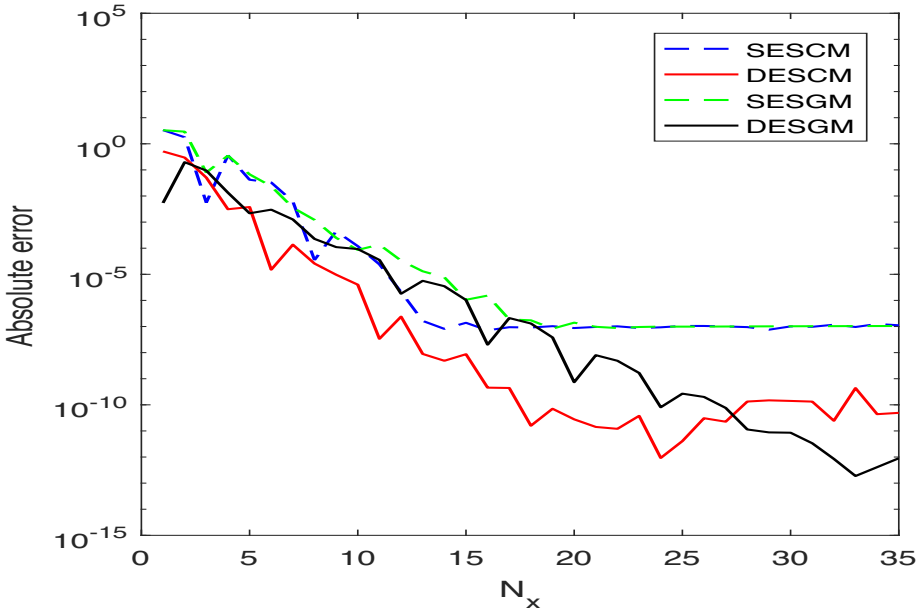

Figure 3.4: Computation of the anharmonic Coulombic potential $V_4(x,y)$. The absolute error for the energy ground level E_0 corresponding to $V_4(x,y)$.

General potentials with randomly selected coefficients

In order to confirm the efficacy of the proposed approaches, we will now apply the method to more complicated potentials of the following form:

$$V_6(x,y) = \sum_{i=-p}^{q} a_i x^i + \sum_{i=-p}^{q} a_i y^i \tag{3.1.36}$$

with $p = 3$ and $q = 8$. The ten coefficients a_i, $i = -2,\ldots,7$ are selected randomly such that $-5 \le a_i \le 5$, and we choose $a_{-3} = 10$ and $a_8 = 5$ to guarantee that the two parameters are positive.

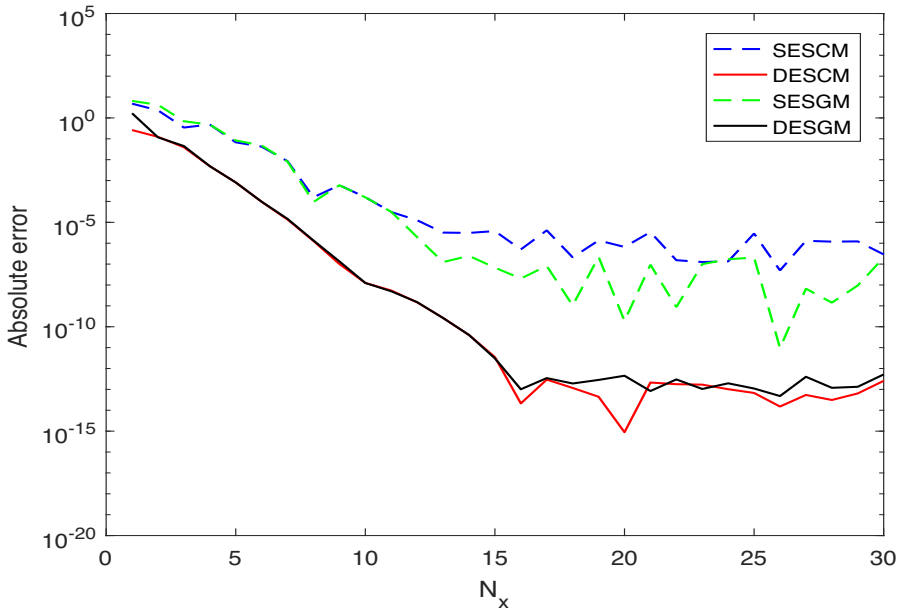

Figure 3.5: Computation of the perturbed anharmonic Coulombic potential $V_5(x, y)$. The absolute error for the energy ground level E_0 corresponding to $V_5(x, y)$.

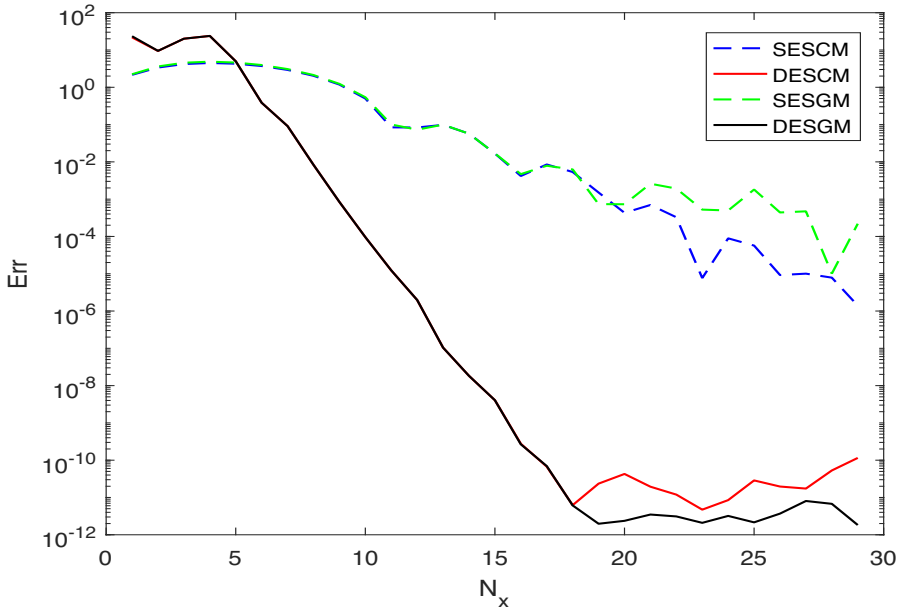

Figure 3.6: Computation of the perturbed anharmonic Coulombic potential $V_6(x, y)$. The absolute error for the energy ground level E_0 corresponding to $V_6(x, y)$.

Since the exact values of the ground state energy E_0 and of the first excited energy E_1 are unknown, we use the following equation as an approximation for the absolute error:

$$Err\ (N_x)\ =\ |\epsilon(N_x)\ -\ \epsilon(N_x-1)|. \tag{3.1.37}$$

In this example, we set:

$$M_x = max\left(\left\lfloor \frac{\gamma_R}{\gamma_L}N_x + \frac{log(\beta_R/\beta_L)}{\gamma_L h_x}\right\rfloor, 0\right), \tag{3.1.38}$$

for an arbitrary N_x, where $\beta_L = \frac{2}{\sqrt{10}}$, $\beta_R = \frac{1}{\sqrt{5}}$, $\gamma_L = \frac{1}{2}$, $\gamma_R = 5$ and $N_y = N_x$, $M_y = M_x$.

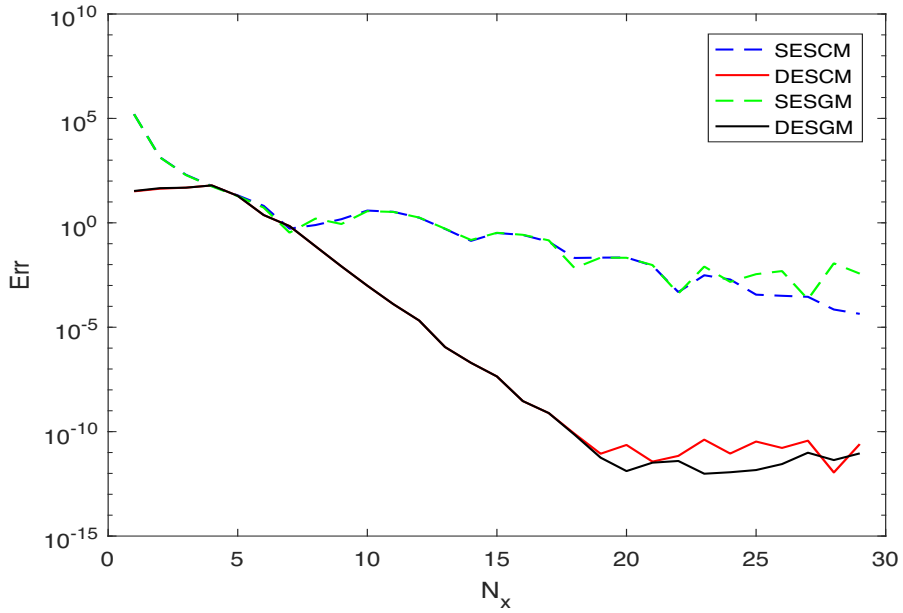

Figure 3.7: Computation of the perturbed anharmonic Coulombic potential $V_6(x, y)$. The absolute error for the first excited energy E_1 corresponding to $V_6(x, y)$.

The numerical results clearly show the high efficiency and accuracy of the DESCM and DESGM for solving the two-dimensional Schrödinger equation with separable potential functions.

3.2 Three-dimensional Schrödinger equation

The discretized Schrödinger equation in three variables is considered in this section. Sinc numerical methods are applied for the resolution and double exponential transformations are used to improve the convergence of the solution. The basis functions and the grid points we use are extensions of those used in the one and two variables studies.

3.2.1 Double exponential Sinc collocation method

The approximate function

$$\psi_A(x,y,z) = \sum_{k=-M_z}^{N_z} \sum_{j=-M_y}^{N_y} \sum_{i=-M_x}^{N_x} u_{ijk} S_{ijk}(x,y,z), \tag{3.2.1}$$

where

$$\begin{aligned} S_{ijk}(x,y,z) &\equiv S_i(x)\, S_j(y)\, S_k(z) \\ &= S(i,h_x)\circ\phi_x(x)\, S(j,h_y)\circ\phi_y(y)\, S(k,h_z)\circ\phi_z(z), \end{aligned} \tag{3.2.2}$$

provides an accurate approximation of the exact solution ψ of the Schrödinger equation.

The method requires substituting the above approximation into the equation (3.0.1) and evaluating the resulting system at the grid nodes:

$$(x_l, y_m, z_n) = \left(\phi_x^{-1}(lh_x), \phi_y^{-1}(mh_y), \phi_z^{-1}(nh_z)\right), \tag{3.2.3}$$

for $l = -M_x, \ldots, N_x$, $m = -M_y, \ldots, N_y$ and $n = -M_z, \ldots, N_z$, with h_x, h_y and h_z the step sizes. That is,

$$-\left(\frac{\partial^2\psi_A}{\partial x^2} + \frac{\partial^2\psi_A}{\partial y^2} + \frac{\partial^2\psi_A}{\partial z^2}\right)(x_l,y_m,z_n) + V(x_l,y_m,z_n)\,\psi_A(x_l,y_m,z_n) = \epsilon\,\psi_A(x_l,y_m,z_n), \tag{3.2.4}$$

where ϵ is an approximation of the energy eigenvalue E. It follows that:

$$\begin{aligned} &-\sum_{k=-M_z}^{N_z}\sum_{j=-M_y}^{N_y}\sum_{i=-M_x}^{N_x} u_{ijk}\left(\frac{\partial^2 S_{ijk}}{\partial x^2} + \frac{\partial^2 S_{ijk}}{\partial y^2} + \frac{\partial^2 S_{ijk}}{\partial z^2}\right)(x_l,y_m,z_n) \\ &\qquad + V(x_l,y_m,z_n)\sum_{k=-M_z}^{N_z}\sum_{j=-M_y}^{N_y}\sum_{i=-M_x}^{N_x} u_{ijk} S_{ijk}(x_l,y_m,z_n) \\ &\qquad\qquad = \epsilon \sum_{k=-M_z}^{N_z}\sum_{j=-M_y}^{N_y}\sum_{i=-M_x}^{N_x} u_{ijk} S_{ijk}(x_l,y_m,z_n), \end{aligned} \tag{3.2.5}$$

which can be written as:

$$\begin{aligned} &-\sum_{k=-M_z}^{N_z}\sum_{j=-M_y}^{N_y}\sum_{i=-M_x}^{N_x} u_{ijk}\Big(S_i''(x_l)S_j(y_m)S_k(z_n) + S_i(x_l)S_j''(y_m)S_k(z_n) + S_i(x_l)S_j(y_m)S_k''(z_n)\Big) \\ &\qquad + V(x_l,y_m,z_n)\sum_{k=-M_z}^{N_z}\sum_{j=-M_y}^{N_y}\sum_{i=-M_x}^{N_x} u_{ijk} S_i(x_l)S_j(y_m)S_k(z_n) \\ &\qquad\qquad = \epsilon \sum_{k=-M_z}^{N_z}\sum_{j=-M_y}^{N_y}\sum_{i=-M_x}^{N_x} u_{ijk} S_i(x_l)S_j(y_m)S_k(z_n). \end{aligned} \tag{3.2.6}$$

Thus, we get:

$$-\sum_{k=-M_z}^{N_z}\sum_{j=-M_y}^{N_y}\sum_{i=-M_x}^{N_x} u_{ijk}\left[\left(\frac{1}{h_x}\phi_x''(x_l)\delta_{il}^{(1)}+\frac{1}{h_x^2}\phi_x'(x_l)^2\delta_{il}^{(2)}\right)\delta_{jm}^{(0)}\delta_{kn}^{(0)}\right.$$
$$+\left(\frac{1}{h_y}\phi_y''(y_m)\delta_{jm}^{(1)}+\frac{1}{h_y^2}\phi_y'(y_m)^2\delta_{jm}^{(2)}\right)\delta_{il}^{(0)}\delta_{kn}^{(0)}+\left(\frac{1}{h_z}\phi_z''(z_n)\delta_{kn}^{(1)}+\frac{1}{h_z^2}\phi_z'(z_n)^2\delta_{kn}^{(2)}\right)\delta_{il}^{(0)}\delta_{jm}^{(0)}\Big]$$
$$+V(x_l,y_m,z_n)\sum_{k=-M_z}^{N_z}\sum_{j=-M_y}^{N_y}\sum_{i=-M_x}^{N_x} u_{ijk}\delta_{il}^{(0)}\delta_{jm}^{(0)}\delta_{kn}^{(0)}=\epsilon\sum_{k=-M_z}^{N_z}\sum_{j=-M_y}^{N_y}\sum_{i=-M_x}^{N_x} u_{ijk}\delta_{il}^{(0)}\delta_{jm}^{(0)}\delta_{kn}^{(0)}. \tag{3.2.7}$$

In the matrix form, using (mat) operator, this system can be expressed as follows:

$$-\Big[\Big(I_{m_y}\otimes A^\top+B^\top\otimes I_{m_x}\Big)\mathrm{mat}(U)+\mathrm{mat}(U)\,C\Big]+\mathrm{mat}(\mathcal{V})\circ\mathrm{mat}(U)=\epsilon\,\mathrm{mat}(U) \tag{3.2.8}$$

with

$$A=\frac{1}{h_x}\cdot*\left(I_{m_x}^{(1)}\,D\left(\phi_x''\right)\right)+\frac{1}{h_x^2}\cdot*\left(I_{m_x}^{(2)}\,D\left(\left(\phi_x'\right)^2\right)\right), \tag{3.2.9}$$

$$B=\frac{1}{h_y}\cdot*\left(I_{m_y}^{(1)}\,D\left(\phi_y''\right)\right)+\frac{1}{h_y^2}\cdot*\left(I_{m_y}^{(2)}\,D\left(\left(\phi_y'\right)^2\right)\right), \tag{3.2.10}$$

$$C=\frac{1}{h_z}\cdot*\left(I_{m_z}^{(1)}\,D\left(\phi_z''\right)\right)+\frac{1}{h_z^2}\cdot*\left(I_{m_z}^{(2)}\,D\left(\left(\phi_z'\right)^2\right)\right), \tag{3.2.11}$$

where ".∗" denotes the component-wise multiplication of a constant by a matrix, $m_x=M_x+N_x+1$, $m_y=M_y+N_y+1$ and $m_z=M_z+N_z+1$. $I_{m_x}^{(p)}$, $I_{m_y}^{(p)}$ and $I_{m_z}^{(p)}$ are the Toeplitz matrices whose elements are $\delta_{il}^{(p)}$, $\delta_{jm}^{(p)}$ and $\delta_{kn}^{(p)}$ for $i,l=-M_x,\ldots,N_x$, $j,m=-M_y,\ldots,N_y$, $k,n=-M_z,\ldots,N_z$ and $p=0,1,2$. Furthermore, $D(\phi_x)$, $D(\phi_y)$ and $D(\phi_z)$ are the diagonal matrices whose diagonal entries are $\phi_x(x_{-M_x}),\ldots,\phi_x(x_{N_x})$, $\phi_y(y_{-M_y}),\ldots,\phi_y(y_{N_y})$ and $\phi_z(z_{-M_z}),\ldots,\phi_z(z_{N_z})$, respectively. $\mathrm{mat}(U)$ and $\mathrm{mat}(\mathcal{V})$ are the $m_xm_y\times m_z$ matrices given, respectively, by $\mathrm{mat}(U)=[\mathrm{vec}\,(u_{i,j,-M_z}),\ldots,\mathrm{vec}\,(u_{i,j,N_z})]$ and $\mathrm{mat}(\mathcal{V})=[\mathrm{vec}\,(V_{i,j,-M_z}),\ldots,\mathrm{vec}\,(V_{i,j,N_z})]$, where $V_{i,j,k}=V(x_i,y_j,z_k)$.

Using the matrix vectorization operator (vec) and the Kronecker product notation ($\otimes$), we obtain:

$$\Big[-\Big(I_{m_z}\otimes\Big(I_{m_y}\otimes A^\top+B^\top\otimes I_{m_x}\Big)+C^\top\otimes I_{m_xm_y}\Big)+D(\mathcal{V})\Big]\mathrm{vec}(U)=\epsilon\,\mathrm{vec}(U). \tag{3.2.12}$$

That is,

$$\Big[-\Big(I_{m_z}\otimes I_{m_y}\otimes A^\top+I_{m_z}\otimes B^\top\otimes I_{m_x}+C^\top\otimes I_{m_y}\otimes I_{m_x}\Big)+D(\mathcal{V})\Big]\mathrm{vec}(U)=\epsilon\,\mathrm{vec}(U), \tag{3.2.13}$$

where $\mathrm{vec}(U)=\mathrm{vec}(\mathrm{mat}(U))$ and $D(\mathcal{V})=diag(\mathrm{vec}(\mathrm{mat}(\mathcal{V})))$.

Particularly, when the potential function is separable, i.e. $V(x,y,z)=V_x(x)+V_y(y)+V_z(z)$, the system (3.2.13) takes the following form:

$$\Big[I_{m_z}\otimes I_{m_y}\otimes\Big(-A^\top+D(V_x)\Big)+I_{m_z}\otimes\Big(-B^\top+D(V_y)\Big)\otimes I_{m_x}$$
$$+\Big(-C^\top+D(V_z)\Big)\otimes I_{m_y}\otimes I_{m_x}\Big]\mathrm{vec}(U)=\epsilon\,\mathrm{vec}(U). \tag{3.2.14}$$

We take advantage from the particular form of the system matrix (3.2.14) which transforms the resolution of one $m_x m_y m_z \times m_x m_y m_z$ system into the resolution of three $m_x \times m_x$, $m_y \times m_y$ and $m_z \times m_z$ systems and leads to reduce the computing time and cost:

$$\Big(- A^\top + D(V_x)\Big) w_1 = \epsilon_1 \, w_1, \tag{3.2.15}$$

$$\Big(- B^\top + D(V_y)\Big) w_2 = \epsilon_2 \, w_2, \tag{3.2.16}$$

$$\Big(- C^\top + D(V_z)\Big) w_3 = \epsilon_3 \, w_3, \tag{3.2.17}$$

where $\epsilon = \epsilon_1 + \epsilon_2 + \epsilon_3$ is the approximation of the exact energy eigenvalue E and $\mathrm{vec}(U) = w_3 \otimes w_2 \otimes w_1$ is the associated eigenfunction.

3.2.2 Double exponential Sinc-Galerkin method

In this section, unless otherwise mentioned, we keep the same notations used previously.

We apply the DESGM for the three-dimensional Schrödinger equation and we obtain the discrete system for the unknown coefficients by orthogonalizing the residual with respect to the inner product:

$$(f, g) = \iiint_\Omega f(x, y, z) g(x, y, z) u(x) v(y) w(z) \, dx \, dy \, dz, \tag{3.2.18}$$

where $u(x)v(y)w(z)$ serves as a weight function. That is:

$$(R, S_{lmn}) = 0, \tag{3.2.19}$$

where

$$R(x, y, z) = -\Delta\psi(x, y, z) + (V(x, y, z) - E) \, \psi(x, y, z), \tag{3.2.20}$$

and

$$S_{lmn}(x, y, z) = S_l(x) S_m(y) S_n(z), \tag{3.2.21}$$

with $-M_x \leq l \leq N_x$, $-M_y \leq m \leq N_y$, $-M_z \leq n \leq N_z$ and h_x, h_y, h_z the step sizes. More precisely, we have:

$$\begin{aligned} \iiint_\Omega \Big(- \Delta\psi(x, y, z) + V(x, y, z)\psi(x, y, z)\Big) S_l(x) S_m(y) S_n(z) u(x) v(y) w(z) dx \, dy \, dz \\ = E \iiint_\Omega \psi(x, y, z) S_l(x) S_m(y) S_n(z) u(x) v(y) w(z) \, dx \, dy \, dz. \end{aligned} \tag{3.2.22}$$

The derivatives of ψ are eliminated using Green's identity. Thus,

$$\begin{aligned} - \iiint_\Omega \psi(x, y, z) \Delta \left(S_l(x) S_m(y) S_n(z) u(x) v(y) w(z) \right) \, dx \, dy \, dz + b_T \\ + \iiint_\Omega V(x, y, z) \psi(x, y, z) S_l(x) S_m(y) S_n(z) u(x) v(y) w(z) \, dx \, dy \, dz \\ = E \iiint_\Omega \psi(x, y, z) S_l(x) S_m(y) S_n(z) u(x) v(y) w(z) \, dx \, dy \, dz, \end{aligned} \tag{3.2.23}$$

where the boundary term b_T is defined by:

$$b_T = \iint_{\partial\Omega} \Big[\psi(x,y,z)\frac{\partial}{\partial n}\Big((S_l u)(x)(S_m v)(y)(S_n w)(z)\Big) - \frac{\partial \psi}{\partial n}(x,y,z)(S_l u)(x)(S_m v)(y)(S_n w)(z)\Big] dS, \tag{3.2.24}$$

and n is the outward normal direction to Ω. Supposing the boundary term b_T vanishes, we get:

$$\begin{aligned}
&-\iiint_\Omega \psi(x,y,z)\Big(S_l(x)u(x)\Big)'' S_m(y)v(y)S_n(z)w(z)\,dx\,dy\,dz \\
&\quad -\iiint_\Omega \psi(x,y,z)S_l(x)u(x)\Big(S_m(y)v(y)\Big)'' S_n(z)w(z)\,dx\,dy\,dz \\
&\quad\quad -\iiint_\Omega \psi(x,y,z)S_l(x)u(x)S_m(y)v(y)\Big(S_n(z)w(z)\Big)''\,dx\,dy\,dz \\
&\quad\quad\quad +\iiint_\Omega V(x,y,z)\psi(x,y,z)S_l(x)S_m(y)S_n(z)u(x)v(y)w(z)\,dx\,dy\,dz \\
&\quad\quad\quad\quad = E\iiint_\Omega \psi(x,y,z)S_l(x)S_m(y)S_n(z)u(x)v(y)w(z)\,dx\,dy\,dz.
\end{aligned} \tag{3.2.25}$$

Applying the quadrature rule and removing the error terms yield:

$$\begin{aligned}
&-h_x h_y h_z \frac{v(y_m)}{\phi_y'(y_m)}\frac{w(z_n)}{\phi_z'(z_n)}\sum_{p=-M_x}^{N_x}\psi(x_p,y_m,z_n)\left[\frac{1}{h_x^2}\delta_{lp}^{(2)}\phi_x'(x_p)u(x_p)+\frac{1}{h_x}\delta_{lp}^{(1)}\left(\frac{\phi_x''(x_p)u(x_p)}{\phi_x'(x_p)}+2u'(x_p)\right)\right. \\
&\left.+\delta_{lp}^{(0)}\frac{u''(x_p)}{\phi_x'(x_p)}\right] - h_x h_y h_z \frac{u(x_l)}{\phi_x'(x_l)}\frac{w(z_n)}{\phi_z'(z_n)}\sum_{q=-M_y}^{N_y}\psi(x_l,y_q,z_n)\left[\frac{1}{h_y^2}\delta_{mq}^{(2)}\phi_y'(y_q)v(y_q)+\frac{1}{h_y}\delta_{mq}^{(1)}\right. \\
&\left.\left(\frac{\phi_y''(y_q)v(y_q)}{\phi_y'(y_q)}+2v'(y_q)\right)+\delta_{mq}^{(0)}\frac{v''(y_q)}{\phi_y'(y_q)}\right] - h_x h_y h_z \frac{u(x_l)}{\phi_x'(x_l)}\frac{v(y_m)}{\phi_y'(y_m)}\sum_{r=-M_z}^{N_z}\psi(x_l,y_m,z_r) \\
&\left[\frac{1}{h_z^2}\delta_{nr}^{(2)}\phi_z'(z_r)w(z_r)+\frac{1}{h_z}\delta_{nr}^{(1)}\left(\frac{\phi_z''(z_r)w(z_r)}{\phi_z'(z_r)}+2w'(z_r)\right)+\delta_{nr}^{(0)}\frac{w''(z_r)}{\phi_z'(z_r)}\right] \\
&+h_x h_y h_z \frac{u(x_l)}{\phi_x'(x_l)}\frac{v(y_m)}{\phi_y'(y_m)}\frac{w(z_n)}{\phi_z'(z_n)}V(x_l,y_m,z_n)\psi(x_l,y_m,z_n) \\
&= h_x h_y h_z E\,\frac{u(x_l)}{\phi_x'(x_l)}\frac{v(y_m)}{\phi_y'(y_m)}\frac{w(z_n)}{\phi_z'(z_n)}\psi(x_l,y_m,z_n).
\end{aligned} \tag{3.2.26}$$

If we divide (3.2.26) by $h_x h_y h_z$ and replace $\psi(x_p,y_q,z_r)$ and E by ψ_{pqr} and ϵ, respectively, we obtain the following discrete system:

$$
\begin{aligned}
&- \frac{v(y_m)}{\phi_y'(y_m)}\frac{w(z_n)}{\phi_z'(z_n)}\sum_{p=-M_x}^{N_x}\psi_{pmn}\left[\frac{1}{h_x^2}\delta_{lp}^{(2)}\phi_x'(x_p)u(x_p)+\frac{1}{h_x}\delta_{lp}^{(1)}\left(\frac{\phi_x''(x_p)u(x_p)}{\phi_x'(x_p)}+2u'(x_p)\right)+\delta_{lp}^{(0)}\frac{u''(x_p)}{\phi_x'(x_p)}\right]\\
&\quad- \frac{u(x_l)}{\phi_x'(x_l)}\frac{w(z_n)}{\phi_z'(z_n)}\sum_{q=-M_y}^{N_y}\psi_{lqn}\left[\frac{1}{h_y^2}\delta_{mq}^{(2)}\phi_y'(y_q)v(y_q)+\frac{1}{h_y}\delta_{mq}^{(1)}\left(\frac{\phi_y''(y_q)v(y_q)}{\phi_y'(y_q)}+2v'(y_q)\right)+\delta_{mq}^{(0)}\frac{v''(y_q)}{\phi_y'(y_q)}\right]\\
&\quad\quad- \frac{u(x_l)}{\phi_x'(x_l)}\frac{v(y_m)}{\phi_y'(y_m)}\sum_{r=-M_z}^{N_z}\psi_{lmr}\left[\frac{1}{h_z^2}\delta_{nr}^{(2)}\phi_z'(z_r)w(z_r)+\frac{1}{h_z}\delta_{nr}^{(1)}\left(\frac{\phi_z''(z_r)w(z_r)}{\phi_z'(z_r)}+2w'(z_r)\right)\right.\\
&\quad\quad\quad\left.+\delta_{nr}^{(0)}\frac{w''(z_r)}{\phi_z'(z_r)}\right]+\frac{u(x_l)}{\phi_x'(x_l)}\frac{v(y_m)}{\phi_y'(y_m)}\frac{w(z_n)}{\phi_z'(z_n)}V(x_l,y_m,z_n)\psi_{lmn}=\epsilon\,\frac{u(x_l)}{\phi_x'(x_l)}\frac{v(y_m)}{\phi_y'(y_m)}\frac{w(z_n)}{\phi_z'(z_n)}\psi_{lmn}.
\end{aligned}
\tag{3.2.27}
$$

Multiplying (3.2.27) by $\phi_x'(x_l)\phi_y'(y_m)\phi_z'(z_n)$, we get:

$$
\begin{aligned}
&-\phi_x'(x_l)v(y_m)w(z_n)\sum_{p=-M_x}^{N_x}\psi_{pmn}\left[\frac{1}{h_x^2}\delta_{lp}^{(2)}\phi_x'(x_p)u(x_p)+\frac{1}{h_x}\delta_{lp}^{(1)}\left(\frac{\phi_x''(x_p)u(x_p)}{\phi_x'(x_p)}+2u'(x_p)\right)\right.\\
&\quad\left.+\delta_{lp}^{(0)}\frac{u''(x_p)}{\phi_x'(x_p)}\right]-u(x_l)\phi_y'(y_m)w(z_n)\sum_{q=-M_y}^{N_y}\psi_{lqn}\left[\frac{1}{h_y^2}\delta_{mq}^{(2)}\phi_y'(y_q)v(y_q)+\frac{1}{h_y}\delta_{mq}^{(1)}\left(\frac{\phi_y''(y_q)v(y_q)}{\phi_y'(y_q)}\right.\right.\\
&\quad\left.\left.+2v'(y_q)\right)+\delta_{mq}^{(0)}\frac{v''(y_q)}{\phi_y'(y_q)}\right]-u(x_l)v(y_m)\phi_z'(z_n)\sum_{r=-M_z}^{N_z}\psi_{lmr}\left[\frac{1}{h_z^2}\delta_{nr}^{(2)}\phi_z'(z_r)w(z_r)+\frac{1}{h_z}\delta_{nr}^{(1)}\right.\\
&\quad\left.\left(\frac{\phi_z''(z_r)w(z_r)}{\phi_z'(z_r)}+2w'(z_r)\right)+\delta_{nr}^{(0)}\frac{w''(z_r)}{\phi_z'(z_r)}\right]+u(x_l)v(y_m)w(z_n)V(x_l,y_m,z_n)\psi_{lmn}\\
&\quad\quad=\epsilon\,u(x_l)v(y_m)w(z_n)\psi_{lmn}.
\end{aligned}
\tag{3.2.28}
$$

Using the matrix operator (mat) and the Kronecker product notation ($\otimes$), (3.2.28) can be expressed in a matrix form as follows:

$$
-\left[\left(I_{m_y}^{(0)}\otimes A+B^\top\otimes I_{m_x}^{(0)}\right)\mathrm{mat}(U)+\mathrm{mat}(U)\,C^\top\right]+\mathrm{mat}\,(\mathcal{V})\circ\mathrm{mat}(U)=\epsilon\,\mathrm{mat}(U),
\tag{3.2.29}
$$

where

$$
A=D(\phi_x')\,\mathcal{A}(u)\,D\left(\frac{1}{u}\right),\quad B=D\left(\frac{1}{v}\right)\,\mathcal{B}(v)^\top\,D(\phi_y')\quad\text{and}\quad C=D\left(\frac{1}{w}\right)\,\mathcal{C}(w)^\top\,D(\phi_z')
\tag{3.2.30}
$$

with

$$
\mathcal{A}(u)=\frac{1}{h_x^2}.*\left(I_{m_x}^{(2)}D(\phi_x'u)\right)+\frac{1}{h_x}.*\left(I_{m_x}^{(1)}D\left(\frac{\phi_x''u}{\phi_x'}+2u'\right)\right)+I_{m_x}^{(0)}D\left(\frac{u''}{\phi_x'}\right)
\tag{3.2.31}
$$

$$
\mathcal{B}(v)=\frac{1}{h_y^2}.*\left(I_{m_y}^{(2)}D(\phi_y'v)\right)+\frac{1}{h_y}.*\left(I_{m_y}^{(1)}D\left(\frac{\phi_y''v}{\phi_y'}+2v'\right)\right)+I_{m_y}^{(0)}D\left(\frac{v''}{\phi_y'}\right)
\tag{3.2.32}
$$

$$
\mathcal{C}(w)=\frac{1}{h_z^2}.*\left(I_{m_z}^{(2)}D(\phi_z'w)\right)+\frac{1}{h_z}.*\left(I_{m_z}^{(1)}D\left(\frac{\phi_z''w}{\phi_z'}+2w'\right)\right)+I_{m_z}^{(0)}D\left(\frac{w''}{\phi_z'}\right)
\tag{3.2.33}
$$

$$
\mathrm{mat}(U)=\left(D(v)\otimes D(u)\right)\mathrm{mat}(\Psi)\,D(w),
\tag{3.2.34}
$$

where $\mathrm{mat}(\Psi)$ is the $m_x m_y \times m_z$ matrix given by $\mathrm{mat}(\Psi) = [\mathrm{vec}(\psi_{i,j,-M_z}), \ldots, \mathrm{vec}(\psi_{i,j,N_z})]$ and $D(u)$, $D(v)$ and $D(w)$ are the diagonal matrices whose diagonal entries are $u(x_{-M_x}), \ldots, u(x_{N_x})$, $v(y_{-M_y}), \ldots, v(y_{N_y})$ and $w(z_{-M_z}), \ldots, w(z_{N_z})$, respectively.

After solving (3.2.29) for U, we can readily get Ψ by the equation:

$$vec(\Psi) = \left[D\left(\frac{1}{w}\right) \otimes D\left(\frac{1}{v}\right) \otimes D\left(\frac{1}{u}\right)\right] vec(U), \tag{3.2.35}$$

taking into account the operators relation $\mathrm{vec}(U) = \mathrm{vec}(\mathrm{mat}(U))$. Finally, in order to eliminate the boundary term b_T, we let the weight functions:

$$u(x) = (\phi_x'(x))^{-1}, \quad v(y) = (\phi_y'(y))^{-1} \quad \text{and} \quad w(z) = (\phi_z'(z))^{-1}. \tag{3.2.36}$$

3.2.3 Numerical results

In this section, we attempt to illustrate the efficiency of the double exponential Sinc collocation method (DESCM) and the double exponential Sinc-Galerkin method (DESGM) when applied to solve the three-dimensional time-independent Schrödinger equation. We compare the performance of the DESCM and DESGM with the performance of the SESCM and SESGM, respectively.

We utilize the mesh size stated as:

$$h = \frac{W(\pi d\gamma n/\beta)}{\gamma n}. \tag{3.2.37}$$

Unless otherwise stated, we choose the positive integer N_x arbitrarily and $N_z = N_y = M_y = M_z = M_x = N_x$.

The figures presented below show the absolute error between the exact values of the energy and the approximations obtained using the proposed approaches. For each example, the curve in the left-hand side shows the results obtained using the DESCM and its comparison to the SESCM, and that in the right-hand side shows the results obtained using the DESGM and its comparison to the SESGM.

Example 1: Anharmonic Oscillator potentials

We present numerical results for the energy values corresponding to the anharmonic oscillator potentials. The study domain is chosen as $\Omega = (-\infty, \infty) \times (-\infty, \infty) \times (-\infty, \infty)$. We utilize the following conformal mappings: $\phi_t(t) = \sinh(t)$. We extend some of the potentials presented in the one-dimensional study and which have known analytic solutions for energy levels :

$$V_1(x,y,z) = V_{x1}(x) + V_{y1}(y) + V_{z1}(z), \quad \text{where } V_{t1}(t) = t^2 - 4t^4 + t^6 \Rightarrow E_0 = -2 - 2 - 2 = -6, \tag{3.2.38}$$

$$V_2(x,y,z) = V_{x2}(x) + V_{y2}(y) + V_{z2}(z), \quad \text{where } V_{t2}(t) = 4t^2 - 6t^4 + t^6 \Rightarrow E_1 = -9 - 9 - 9 = -27, \tag{3.2.39}$$

for $t = x, y$ or z.

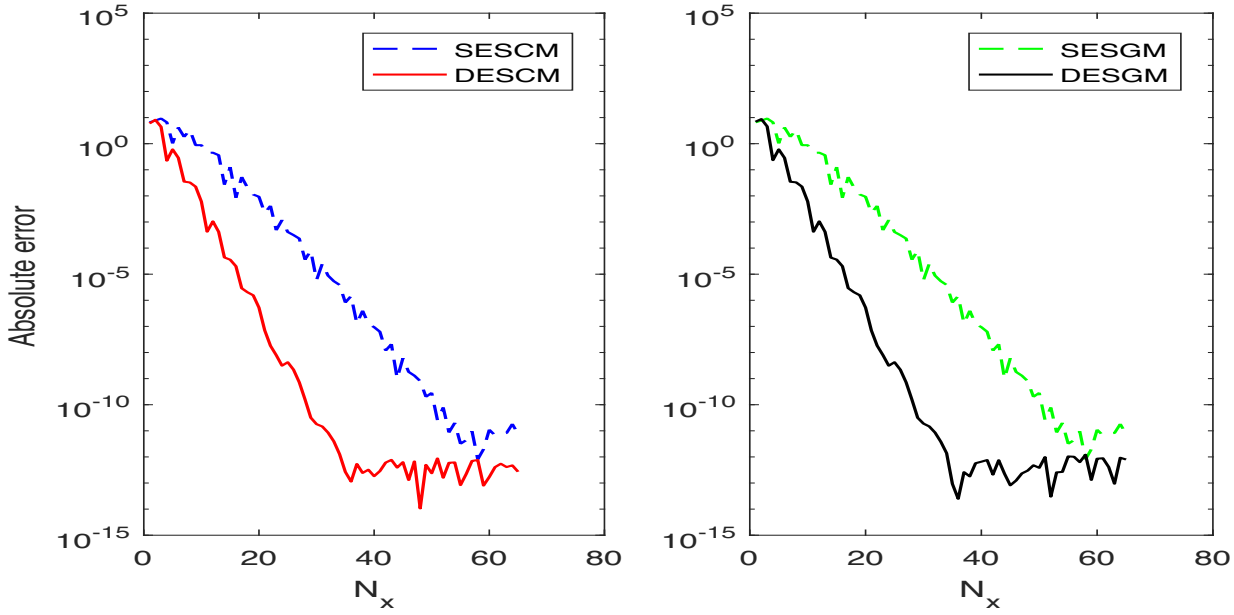

Figure 3.8: Convergence history of DESinc methods and comparison with SESinc methods for E_0 corresponding to $V_1(x, y, z)$.

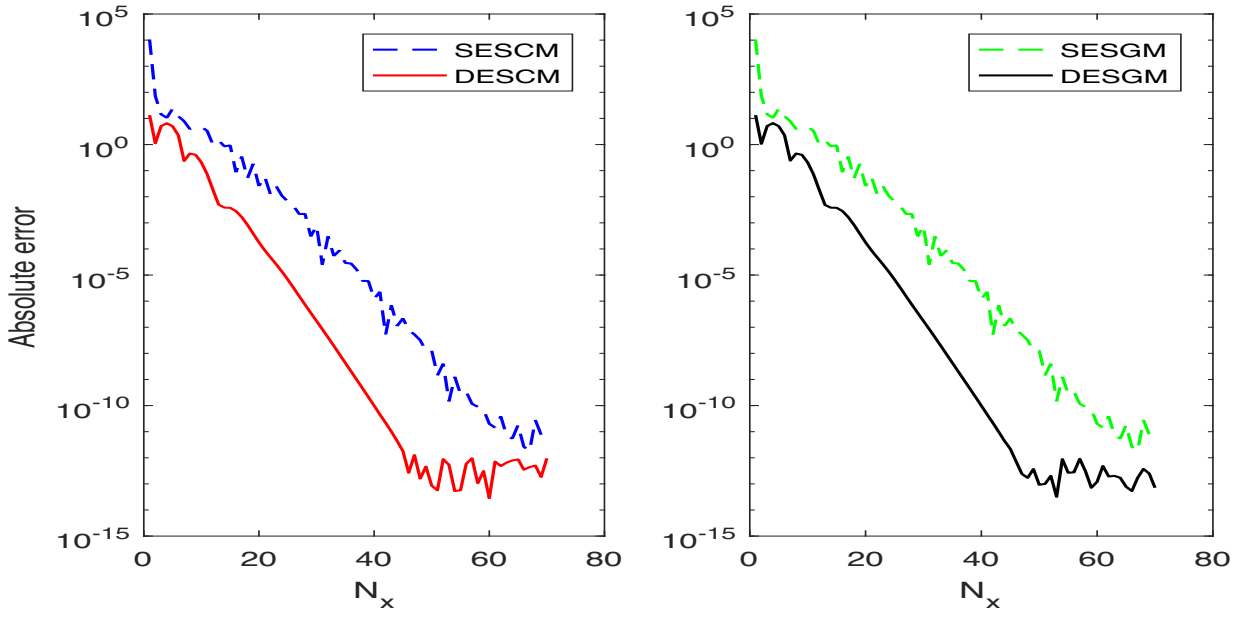

Figure 3.9: Convergence history of DESinc methods and comparison with SESinc methods for E_1 corresponding to $V_2(x, y, z)$.

Figure 3.8 displays the absolute error for the energy ground level E_0 corresponding to the potential $V_1(x, y, z)$ and Figure 3.9 shows the absolute error for the first excited energy level E_1 corresponding to the potential $V_2(x, y, z)$.

Example 2: Anharmonic Coulombic potentials

Consider the three-dimensional Schrödinger equation with the following anharmonic coulombic potentials:

$$V_3(x,y,z) = V_{x3}(x) + V_{y3}(y) + V_{z3}(z), \quad \text{where} \quad V_{t3}(t) = \frac{2}{t^2} + t^2 \quad \Rightarrow \; E_0 = 5+5+5 = 15, \tag{3.2.40}$$

$$V_4(x,y,z) = V_{x4}(x) + V_{y4}(y) + V_{z4}(z), \quad \text{where} \quad V_{t4}(t) = \frac{3}{4t^2} + t^2 \quad \Rightarrow \; E_0 = 4+4+4 = 12. \tag{3.2.41}$$

The domain Ω is selected as $(0,\infty)\times(0,\infty)\times(0,\infty)$. We utilize the following conformal mappings: for $t = x, y$ or z, $\phi_t(t) = \log(\mathrm{e}^{\sinh(t)} + 1)$. For the implementation of the DESCM and the DESGM, we choose: $M_x = max\left(\left\lfloor \frac{\gamma_R}{\gamma_L} N_x + \frac{log(\beta_R/\beta_L)}{\gamma_L h_x} \right\rfloor, 0\right)$, for an arbitrary N_x, where $\beta_L = 1$, $\beta_R = \frac{1}{8}$, $\gamma_L = 1$, $\gamma_R = 2$ and $N_z = N_y = N_x$, $M_z = M_y = M_x$.

Concerning the comparison with the SESinc methods, we implement the SESCM and SESGM using the transformations in Table 1.1.

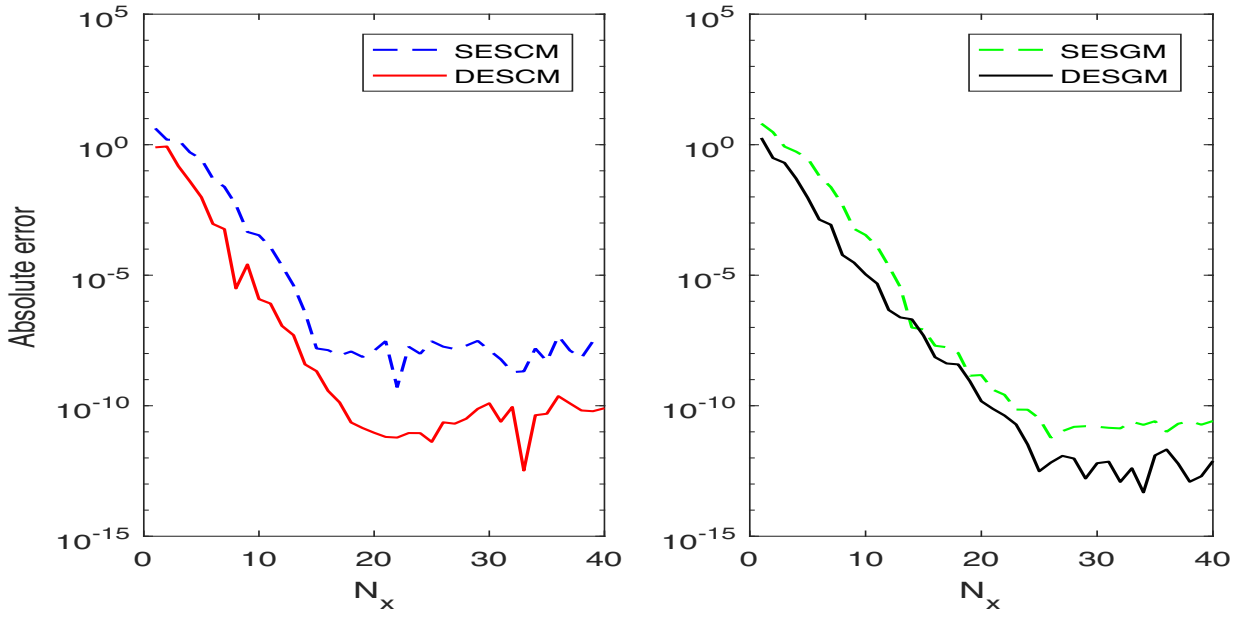

Figure 3.10: Convergence history of DESinc methods and comparison with SESinc methods for E_0 corresponding to $V_3(x, y, z)$.

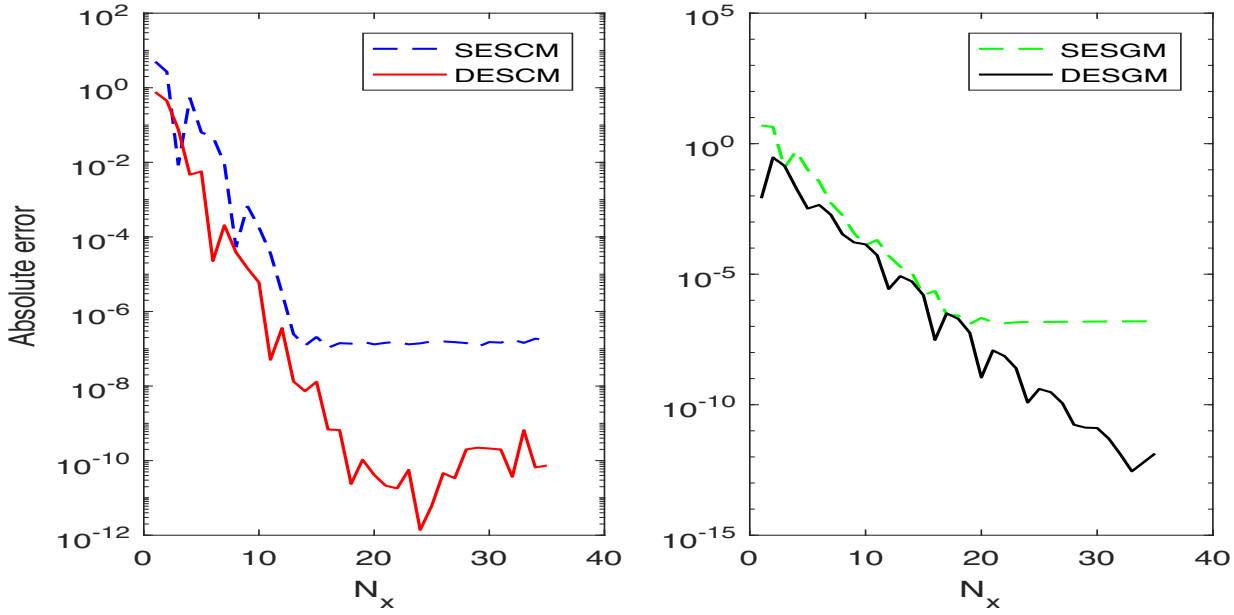

Figure 3.11: Convergence history of DESinc methods and comparison with SESinc methods for E_0 corresponding to $V_4(x, y, z)$.

Figure 3.10 and Figure 3.11 show the convergence history of the DESCM and the DESGM for the potentials $V_3(x, y, z)$ and $V_4(x, y, z)$, respectively, with known analytic solutions for the ground state energy E_0.

Example 3: Perturbed Anharmonic Coulombic potentials

Consider the anharmonic Coulombic potentials with singular perturbation of the form:

$$V_5(x,y,z) = V_{x5}(x)+V_{y5}(y)+V_{z5}(z), \quad \text{where} \quad V_{t5}(t) = t^2+\frac{9}{64t^6} \quad \Rightarrow \quad E_0 = 4+4+4 = 12. \quad (3.2.42)$$

We select the domain Ω as $(0,\infty)\times(0,\infty)\times(0,\infty)$ and we utilize the following conformal mappings: $\phi_t(t) = \mathrm{e}^t$, for $t = x, y$ or z.

Figure 3.12 presents the absolute error between DESinc and SESinc methods approximations and the exact value of the energy ground level E_0 for the potential $V_5(x, y, z)$.

Now, we will try more complicated potentials in the following form:

$$V_6(x,y,z) = V_{x6}(x) + V_{y6}(y) + V_{z6}(z), \qquad \text{where} \qquad V_{t6}(t) = \sum_{i=-p}^{q} a_i t^i \qquad (3.2.43)$$

for $t = x, y$ and z, with $p = 3$ and $q = 8$.

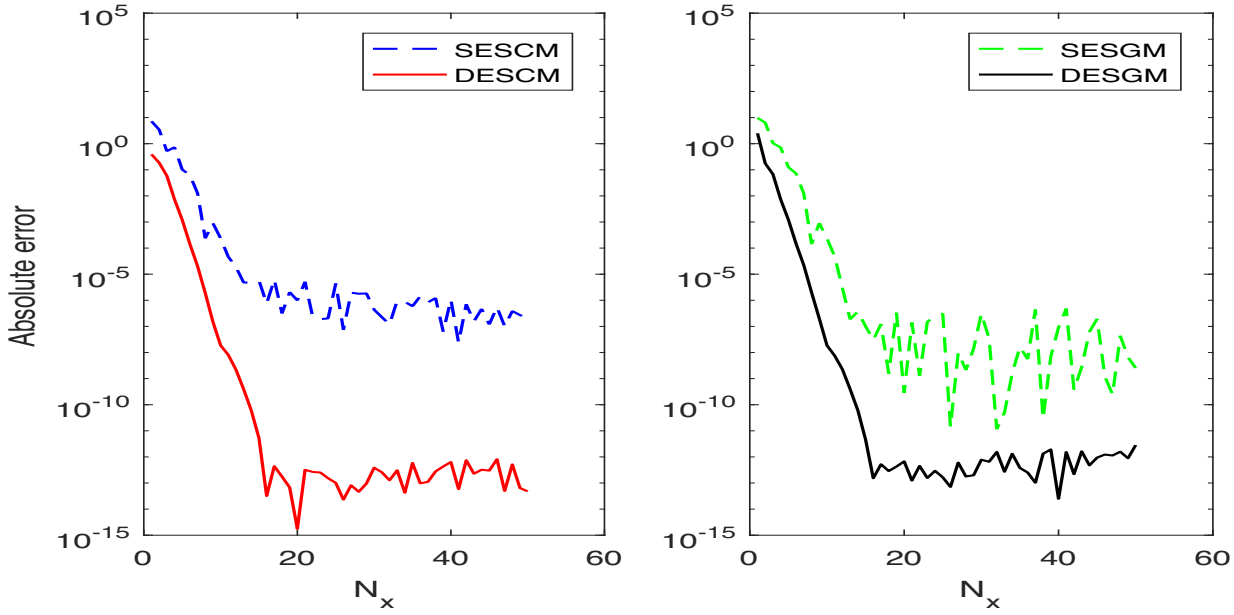

Figure 3.12: Convergence history of DESinc methods and comparison with SESinc methods for E_0 corresponding to $V_5(x, y, z)$.

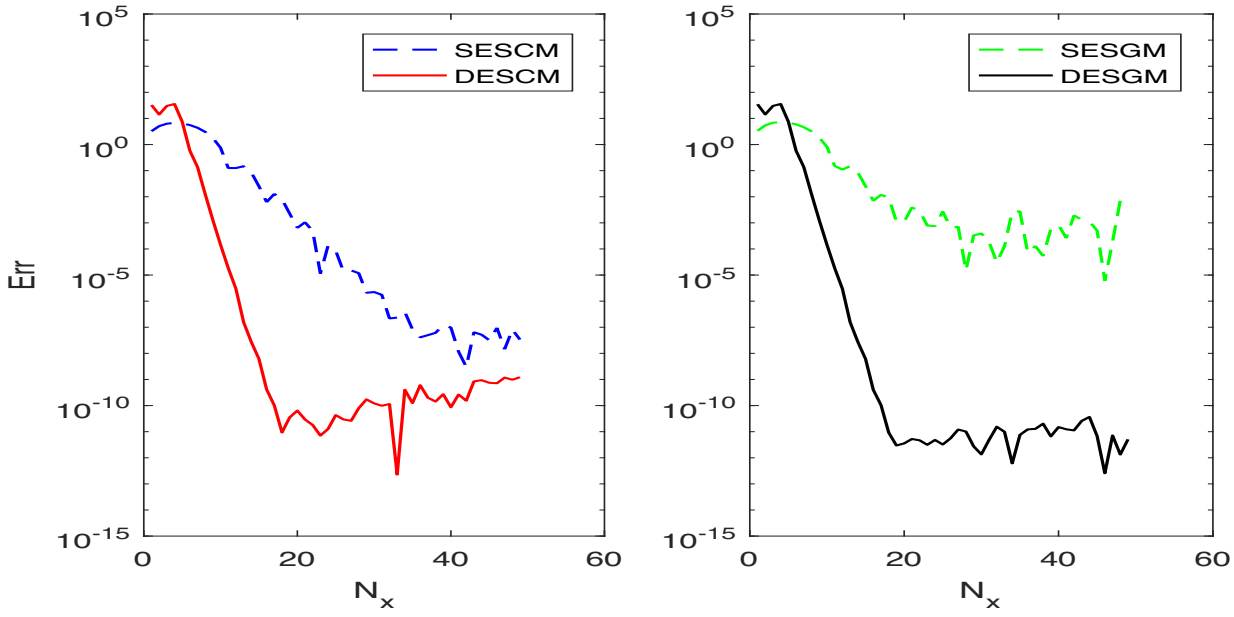

Figure 3.13: Convergence history of DESinc methods and comparison with SESinc methods for E_0 corresponding to $V_6(x, y, z)$.

We pick the 10 coefficients $a_i,\ i = -2, \ldots, 7$ randomly such that $-5 \leq a_i \leq 5$, and we choose $a_{-3} = 10$ and $a_8 = 5$ to guarantee the positivity of these two parameters. For the potential V_6, we pick $M_x = max\left(\left\lfloor \frac{\gamma_R}{\gamma_L} N_x + \frac{log(\beta_R/\beta_L)}{\gamma_L h_x} \right\rfloor, 0\right)$, for an arbitrary N_x, where $\beta_L = \frac{2}{\sqrt{10}}$, $\beta_R = \frac{1}{\sqrt{5}}$, $\gamma_L = \frac{1}{2}$, $\gamma_R = 5$ and $N_z = N_y = N_x$, $M_z = M_y = M_x$.

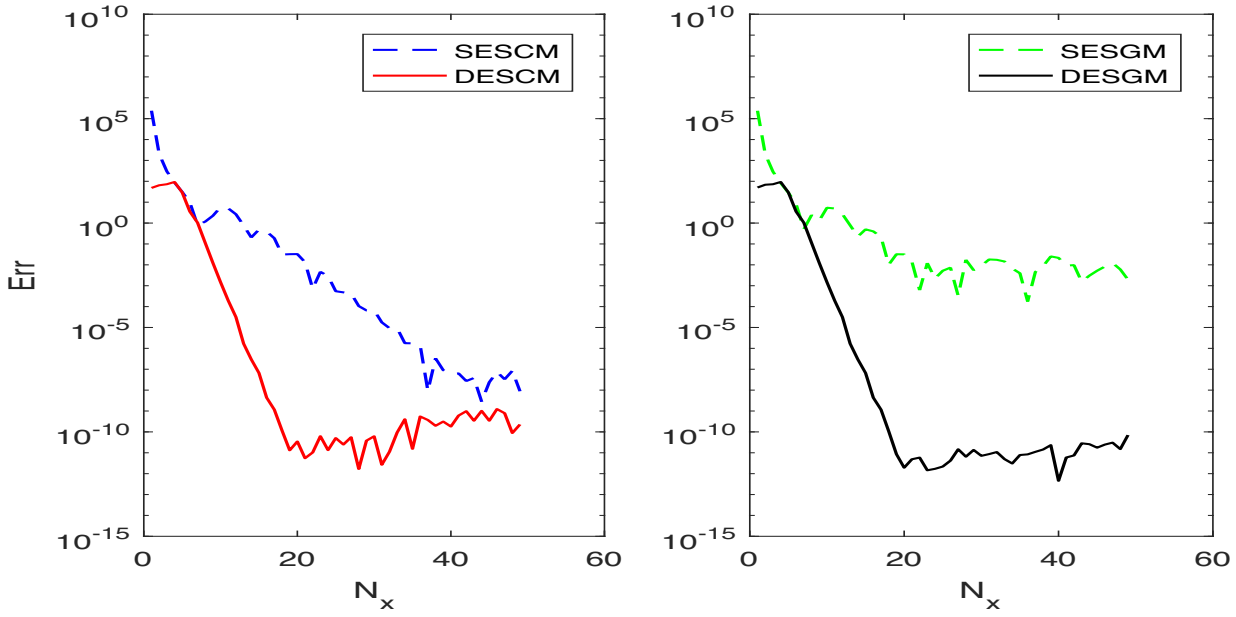

Figure 3.14: Convergence history of DESinc methods and comparison with SESinc methods for E_1 corresponding to $V_6(x, y, z)$.

The Figures 3.13 and 3.14 display the approximation of the absolute error Err for the ground state energy E_0 and the first excited energy E_1, corresponding to $V_6(x, y, z)$, respectively.

Conclusion

In this chapter, we applied the double exponential Sinc methods for the two- and three- dimensional time-independent Schrödinger equation. Numerical experiments illustrate the behavior and the accuracy of the proposed methods. As can be seen in the figures, the DESCM and DESGM perform better than the SESCM and SESGM for high values of N_x. In two and three dimensions, the size of the Hamiltonian matrix grows rapidly as the number of spatial grid points increases which requires more computing time and cost. Here, we benefited from the separable case which reduces the Schrödinger equation to one-dimensional problems and lowers the computational resources.

Chapter 4

Block centrosymmetry

Centrosymmetric matrices arise in several problems in statistics, engineering, communication theory and in the numerical resolution of differential equations [56, 57, 58, 40]. This class of matrices is characterized by its rich eigenstructure which has been widely explored by many authors [59, 36, 37]. The most known results about the centrosymmerty were based on the orthogonal similarity transformations that were useful in block-diagonalizing the centrosymmetric matrices. In this chapter, we extend the centrosymmetric property to a bidimensional space, by working with block matrices as components of the total matrix.

4.1 Centrosymmetric basic properties

In this section, we give some preliminaries [36] about the centrosymmetric matrices.

Definition 4.1.1 *An exchange matrix denoted by J is an $m \times m$ square matrix with ones along the anti-diagonal and zeros everywhere else:*

$$J = \begin{pmatrix} 0 & & 1 \\ & \iddots & \\ 1 & & 0 \end{pmatrix}. \tag{4.1.1}$$

Definition 4.1.2 *Let M be a square matrix of dimension $m \times m$. M is centrosymmetric if and only if it verifies the following property:*

$$MJ = JM, \tag{4.1.2}$$

where J is an exchange matrix of dimension $m \times m$.

The lemmas below will show the internal block structure of centrosymmetric matrices and its usefulness for the computation of the eigenvalues.

Lemma 4.1.3 *If* H *is a square centrosymmetric matrix of dimension* $m \times m$ *where* $m = 2N + 1$, *then* H *can be expressed as:*

$$\mathrm{H} = \begin{pmatrix} S & x & JCJ \\ y & h & yJ \\ C & Jx & JSJ \end{pmatrix}, \tag{4.1.3}$$

where S, C *are matrices of size* $N \times N$, J *is the exchange matrix of size* $N \times N$, x *and* y *are vectors of length* N *and* h *is a scalar.*

Lemma 4.1.4 *Let* H *be a square centrosymmetric matrix as given in Lemma 4.1.3 and let* V *be a square matrix of dimension* $m \times m$ *defined by:*

$$\mathrm{V} = \begin{pmatrix} S - JC & 0 & 0 \\ 0 & h & \sqrt{2}y \\ 0 & \sqrt{2}x & S + JC \end{pmatrix}, \tag{4.1.4}$$

then H *and* V *are orthogonally similar. That is, the matrices* H *and* V *have the same Jordan normal form and thus the same eigenvalue spectrum.*

The above lemmas will be helpful to prove the following theorem that facilitates solving an eigenvalue problem with a centrosymmetric matrix.

Theorem 4.1.5 *Let* H *be a square centrosymmetric matrix as given in Lemma 4.1.3, then the resolution of the eigenvalue problem:*

$$det(\mathrm{H} - \lambda I) = 0 \tag{4.1.5}$$

is equivalent to the resolution of the two smaller eigenvalue problems:

$$det(S - JC - \lambda I) = 0 \quad and \quad det\left(\begin{pmatrix} h & \sqrt{2}y \\ \sqrt{2}x & S + JC \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & I \end{pmatrix}\right) = 0. \tag{4.1.6}$$

4.2 Block centrosymmetry of the system matrix

In the present section, we introduce a property of the matrix resulting from the DESinc methods resolution of the two-dimensional Schrödinger equation with nonseparable potentials. This property will be helpful in the calculation of the eigenvalues of the system matrix and in significantly reducing the computational time and storage.

Definition 4.2.1 *[60] Let* $\mathcal{J}_y$ *denote the parity operator with respect to* y, *defined by:*

$$\mathcal{J}_y f(x, y) = f(x, -y), \tag{4.2.1}$$

where $f(x, y)$ *is a well defined function being acted upon* $\mathcal{J}_y$.

Definition 4.2.2 *[40] An operator $\mathcal{B}$ is said to commute with the operator $\mathcal{J}_y$ if it verifies the following relation:*

$$\mathcal{B}\mathcal{J}_y f(x,y) = \mathcal{J}_y \mathcal{B} f(x,y). \tag{4.2.2}$$

Equivalently, we may say that the commutator between $\mathcal{B}$ and $\mathcal{J}_y$ is zero, that is:

$$[\mathcal{B}, \mathcal{J}_y] = \mathcal{B}\mathcal{J}_y - \mathcal{J}_y\mathcal{B} = 0. \tag{4.2.3}$$

Definition 4.2.3 *A block exchange matrix denoted by $\mathbf{J}$ is an $m^2 \times m^2$ square matrix with identity matrices I of size $m \times m$ along the block anti-diagonal and zeros everywhere else:*

$$\mathbf{J} = \begin{pmatrix} 0 & & I \\ & \iddots & \\ I & & 0 \end{pmatrix}. \tag{4.2.4}$$

Definition 4.2.4 *Let $\mathbf{M}$ be a matrix of dimension $m^2 \times m^2$, with block components of dimension $m \times m$. $\mathbf{M}$ is block centrosymmetric if and only if $\mathbf{M}$ verifies the following property:*

$$\mathbf{MJ} = \mathbf{JM}, \tag{4.2.5}$$

where $\mathbf{J}$ is a block exchange matrix of dimension $m^2 \times m^2$.

The resolution of the two-dimensional Schrödinger equation using DESinc methods results in the discretized systems (3.1.9) and (3.1.24) whose matrices have the following form:

$$\mathbf{M} = -\left(I_m^{(0)} \otimes A + B \otimes I_m^{(0)}\right) + \mathcal{D}(\mathcal{V}), \tag{4.2.6}$$

assuming that $M_y = M_x = N_x = N_y = N$ and $m_x = m_y = m = 2N + 1$.

Given some particular parity assumptions on the functions $V(x,y)$ and $\phi_y(y)$, the result below will show the connection between the Schrödinger operator and the resulting matrix approximation.

Theorem 4.2.5 *[40] Let $\mathcal{H}$ denote the operator of the two-dimensional Schrödinger equation given by:*

$$\mathcal{H} = -\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right) + V(x,y). \tag{4.2.7}$$

If the commutator $[\mathcal{H}, \mathcal{J}_y] = 0$, where $\mathcal{J}_y$ is the parity operator with respect to y, and if the DE transformation $\phi_y(y)$ is an odd function, then the matrix $\mathbf{M}$ is block centrosymmetric.

Remark 4.2.6 *The commutator $[\mathcal{H}, \mathcal{J}_y] = 0$ if and only if the potential function $V(x,y)$ verifies the following property:*

$$V(x,-y) = V(x,y). \tag{4.2.8}$$

Remark 4.2.7 *If the transformations ϕ_x and ϕ_y are both odd functions, then the matrix $\mathbf{M}$ is centrosymmetric.*

The lemmas below will show the internal block structure of block centrosymmetric matrices and its convenience to facilitate the computation of the eigenvalues.

Lemma 4.2.8 *If $\mathbf{H}$ is a square block centrosymmetric matrix of dimension $m^2 \times m^2$ where $m = 2N + 1$, then $\mathbf{H}$ can be expressed as:*

$$\mathbf{H} = \begin{pmatrix} \mathbf{S} & \mathbf{X} & \mathbf{JCJ} \\ \mathbf{Y} & H & \mathbf{YJ} \\ \mathbf{C} & \mathbf{JX} & \mathbf{JSJ} \end{pmatrix}, \tag{4.2.9}$$

where $\mathbf{S}, \mathbf{C}$ are matrices of size $Nm \times Nm$, $\mathbf{J}$ is the block exchange matrix of size $Nm \times Nm$, $\mathbf{X}$ is a matrix of size $Nm \times m$, $\mathbf{Y}$ is a matrix of size $m \times Nm$ and H is a matrix of size $m \times m$.

Lemma 4.2.9 *Let $\mathbf{H}$ be a square block centrosymmetric matrix as given in Lemma 4.2.8 and let $\mathbf{V}$ be a square matrix of dimension $m^2 \times m^2$ defined by:*

$$\mathbf{V} = \begin{pmatrix} \mathbf{S} - \mathbf{JC} & 0 & 0 \\ 0 & H & \sqrt{2}\mathbf{Y} \\ 0 & \sqrt{2}\mathbf{X} & \mathbf{S} + \mathbf{JC} \end{pmatrix}, \tag{4.2.10}$$

then $\mathbf{H}$ and $\mathbf{V}$ are orthogonally similar. That is, the matrices $\mathbf{H}$ and $\mathbf{V}$ have the same Jordan normal form and thus the same eigenvalue spectrum.

The above lemmas will be useful to prove the following theorem that helps solving an eigenvalue problem with a block centrosymmetric matrix.

Theorem 4.2.10 *Let $\mathbf{H}$ be a square block centrosymmetric matrix as given in Lemma 4.2.8, then solving the eigenvalue problem:*

$$det(\mathbf{H} - \lambda\mathbf{I}) = 0 \tag{4.2.11}$$

is equivalent to solving the two smaller eigenvalue problems:

$$det(\mathbf{S} - \mathbf{JC} - \lambda\mathbf{I}) = 0 \quad and \quad det\left(\begin{pmatrix} H & \sqrt{2}\mathbf{Y} \\ \sqrt{2}\mathbf{X} & \mathbf{S} + \mathbf{JC} \end{pmatrix} - \lambda \begin{pmatrix} I & 0 \\ 0 & \mathbf{I} \end{pmatrix}\right) = 0. \tag{4.2.12}$$

In this section, we presented some useful results that give advantage in solving two eigensystems of dimensions $Nm \times Nm$ and $(N + 1)m \times (N + 1)m$ rather than one eigensystem of dimension $m^2 \times m^2$.

4.3 Numerical results

We consider the two-dimensional time-independent Schrödinger equation with nonseparable potential functions.

4.3.1 Nonseparable potentials

The choice of these potentials is based on the examples presented in [51, 52]. We display numerical results for the energy values corresponding to different nonseparable potentials. The domain is $\Omega = (-\infty, \infty) \times (-\infty, \infty)$, and we used the double exponential conformal mapping $\sinh(t)$.
We first consider the Schrödinger equation with the following potential:

$$V_7(x, y) = (1 + x^2)(1 + y^2), \tag{4.3.1}$$

where the convergence of the corresponding eigenvalue is illustrated in Figure 4.1:

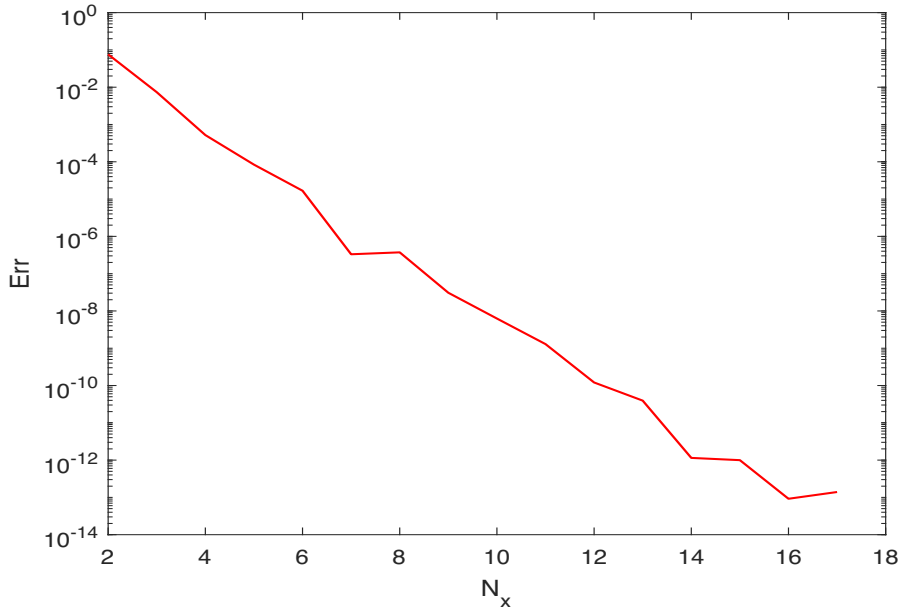

Figure 4.1: Computation of the nonseparable potential $V_7(x, y)$. The absolute error for the energy ground level E_0 corresponding to $V_7(x, y)$.

Figures 4.2 and 4.3 are devoted to the quartic and sextic potentials given, respectively, by:

$$V_8(x, y) = x^2 + y^2 + c(x^4 + 2ax^2y^2 + y^4), \tag{4.3.2}$$

$$V_9(x, y) = x^2 + y^2 + d(x^6 + 3bx^4y^2 + 3bx^2y^4 + y^6), \tag{4.3.3}$$

where $c > 0$, $-1 \leq a \leq 1$, $d > 0$ and $-1/3 \leq b \leq 1$.

Finally, we consider the two-well quartic potential V_{10} whose energy's convergence is given in Figure 4.4:

$$V_{10}(x, y) = -x^2 - y^2 + 0.1(x^2 + y^2)^2. \tag{4.3.4}$$

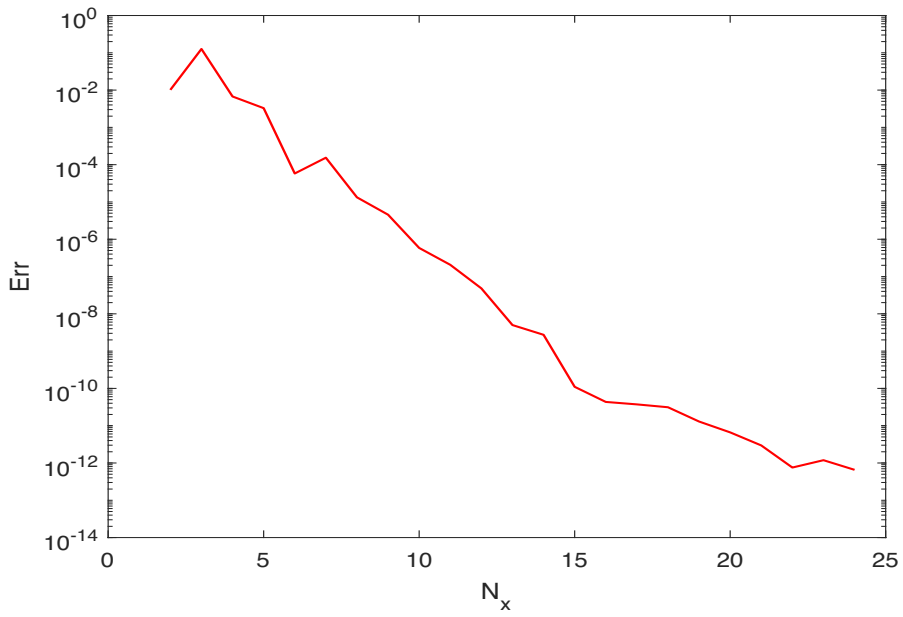

Figure 4.2: Computation of the quartic potential $V_8(x, y)$. The absolute error for the first excited energy E_1 corresponding to $V_8(x, y)$.

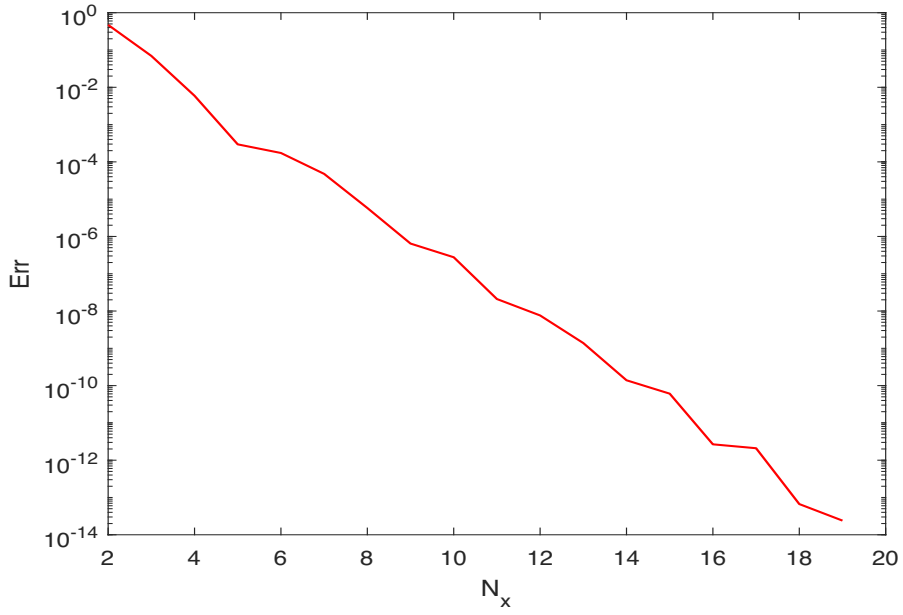

Figure 4.3: Computation of the sextic potential $V_9(x, y)$. The absolute error for the energy ground level E_0 corresponding to $V_9(x, y)$.

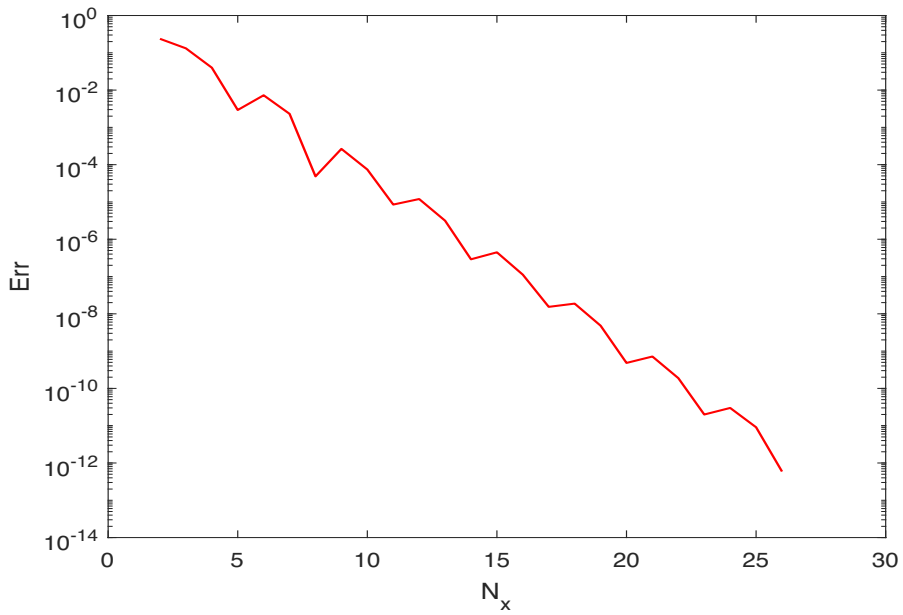

Figure 4.4: Computation of the two-well quartic potential $V_{10}(x, y)$. The absolute error for the energy ground level $E_0 + 2.5$ corresponding to $V_{10}(x, y)$.

The figures showed a great performance of the DESCM for different types of nonseparable potentials. DESGM produced the same accuracy as DESCM. The method was also efficient when applied for the quartic potential with $c = 10^{-3}, 10^3, 10^6, \infty$, $a = 0, 1$ and for the sextic potential with $d = 1, 10^4, \infty$, $b = -1/3, 0, 2/3, 1$. As for the comparison, the results acquired are in agreement with those obtained in [51, 52]. Due to the block centrosymmetric property, the computation time is reduced by a factor 2 for nonseparable potentials.

Conclusion

The numerical results show the efficacy of DESCM and DESGM for solving the Schrödinger equation not only with separable potentials but also with nonseparable potentials. The matrices obtained from the discretization are block centrosymmetric, leading to a substantial reduction in the calculation time.

Part II

Block and Global Lanczos methods for large linear systems

Introduction

This second part of this thesis is devoted for the resolution of large sparse linear systems with the same coefficient matrix and multiple right-hand sides expressed as:

$$Ax^{(i)} = b^{(i)}, \quad i = 1, \ldots, s, \tag{4.3.5}$$

where A is an $n \times n$ nonsingular real matrix. Assuming all the right-hand sides $b^{(i)} \in \mathbb{R}^n$ are available simultaneously, the sequence of linear systems (4.3.5) can then be seen as one linear system written as:

$$AX = B, \tag{4.3.6}$$

where $B = [b^{(1)}, b^{(2)}, \ldots, b^{(s)}]$ and $X = [x^{(1)}, x^{(2)}, \ldots, x^{(s)}]$ are $n \times s$ rectangular matrices with n large and $s << n$. These systems arise in many scientific and engineering applications such as control theory and model reduction [61, 62].

Classically, in order to solve a linear system with small matrices, we use the direct methods such as LU decomposition, but if n is a very large number, then solving the system using standard techniques may be a serious challenge due to large memory and computational requirements. These latter were some of the incentives to utilize iterative methods that have gained popularity dealing with large problems in different areas. Iterative methods are among the sparse matrix techniques that exploit the large number of zero elements of a matrix and their locations, and benefit from the fact that these elements need not be stored which leads to a reduction in the storage requirements. Starting with a given approximate solution, these methods construct a sequence of approximations that converges to the exact solution. Unlike the direct methods, during the execution of the iterative methods, matrix-vector products are used rather than directly operating on the matrix elements, which allow to avoid the repetitive operations on the zero elements of the matrix.

Basic iterative methods such as Jacobi, Gauss-Seidel and SOR, were the first iterative methods applied for the resolution of large linear systems. Basing on the relaxation of the coordinates, they aim to find an approximate solution by modifying one or few components of the approximation until convergence is reached. These techniques are in turn barely used since their convergence is not always guaranteed and it is generally slow, especially for large systems. Although, they can be quite efficient when incorporated with other iterative methods.

Currently, the iterative Krylov methods have been the best known and the most widely used among the existing iterative techniques. They use projection processes for seeking an approximation to the solution in Krylov subspaces either by imposing Petrov-Galerkin conditions, such as Arnoldi method [8], Lanczos method [63], BiConjugate Gradient method (BiCG) [64] and BiConjugate Gradient Stabilized method (BiCGStab) [65], or by utilizing minimization conditions such as

Generalized Minimum Residual method (GMRES) [8] and Quasi-Minimal Residual method (QMR) [64]. Recently, these methods have known development in order to be adjusted to the resolution of linear systems with several right-hand sides. The block Krylov subspace methods are generalizations of the classical Krylov methods, which attempt to find the solution of all the linear systems simultaneously instead of solving each system independently. The block solvers such as Bl-Lanczos method [12], Bl-BiCG method [66], Bl-BiCGStab method [67] and Bl-GMRES method [68], project each column of the initial matrix residual onto a Block Krylov subspace, whereas the global Krylov subspace methods such as Gl-Lanczos method [69], Gl-BiCG method [70], Gl-BiCGStab method [70] and Gl-GMRES method [71], project globally the initial matrix residual onto a matrix Krylov subspace. These global solvers are efficient, comparing to the block solvers, when applied for solving large and sparse low rank right-hand sides Lyapunov and Sylvester matrix equations [72, 71].

In this part, new versions of the block nonsymmetric Lanczos method and the global nonsymmetric Lanczos method are established in order to solve large linear systems with the same matrix and various right-hand sides. These versions are based on the new expressions developed using the Schur complement formulas and the block and global Lanczos process. This new variant of the global Lanczos method is applied for the approximate computation of the transfer function.

Chapter 5

Block Lanczos method for large linear systems with multiple right-hand sides

When dealing with large linear systems with the same coefficient matrix and different right-hand sides, it is preferable to apply a method for all the linear systems simultaneously rather than using a method for each system. The block Lanczos method is among the block Krylov subspace solvers which are generalizations of the standard Krylov subspace methods, for which the matrix system works on a block of vectors instead of a single vector.

Several researchers have used the block Lanczos method for the resolution of large linear problems. In [12], El Guennouni et al. presented new block Lanczos-type algorithms for linear systems with multiple right-hand sides. They have expressed the matrix residual using matrix-valued orthogonal polynomials and derived recurrence formulas for computing these polynomials basing on the Schur complement. In [73], Esghir et al. used the properties of the Schur complements to give a new algorithm for implementing the block Lanczos method in order to approximate the transfer function of linear dynamical systems. These latter were also studied in [74] by applying an extended version of the block Lanczos method with moment matching techniques for model reduction. The authors in [75] proposed a new numerical method, based on the connection between the block Lanczos algorithm and the block Gauss quadrature rules, for the Tikhonov regularization of large linear system of equations with multiple right-hand sides, where the right-hand side matrix is contaminated by an error.

In this chapter, we yearn for incorporating the block nonsymmetric Lanczos algorithm and some Schur complements identities which take advantage from the block partitioning of the matrix to give recursive formulas that help constructing a new variant of the block nonsymmetric Lanczos method.

5.1 Block nonsymmetric Lanczos method

The block Lanczos algorithm is a block analogue of the classical Lanczos algorithm, which uses matrix-matrix products instead of matrix-vector products, and does not need to operate on the

matrix components. It is a projection method that enables the resolution of a large sparse linear system by transforming the system matrix into a smaller banded block tridiagonal one.

5.1.1 Block nonsymmetric Lanczos procedure

Given an $n \times n$ real nonsymmetric and nonsingular matrix A and initial $n \times s$ block vectors V_1 and W_1 such that $W_1^\top V_1 = I_s$, where I_s denotes the $s \times s$ identity matrix. The block nonsymmetric Lanczos process consists in generating basis block vectors $\{V_1, V_2, \ldots, V_m\}$ of the block Krylov subspace:

$$K_m(A, V_1) = \text{span}\left\{V_1, AV_1, \ldots, A^{m-1}V_1\right\} \tag{5.1.1}$$

and basis block vectors $\{W_1, W_2, \ldots, W_m\}$ of the block Krylov subspace:

$$K_m(A^\top, W_1) = \text{span}\left\{W_1, A^\top W_1, \ldots, \left(A^\top\right)^{m-1} W_1\right\}. \tag{5.1.2}$$

The block vectors V_i and W_j, for $i, j = 1, \ldots, m$, are constructed so that:

$$W_i^\top V_j = \begin{cases} I_s, & \text{if } i = j, \\ 0_s & \text{if } i \neq j. \end{cases} \tag{5.1.3}$$

where 0_s denotes the $s \times s$ zero matrix. The algorithm given below executes m steps of the block Lanczos process applied to A:

Algorithm 2 Block nonsymmetric Lanczos process *[76]*

1 Choose two matrices V_1 and W_1 in $\mathbb{R}^{n \times s}$ such that $W_1^\top V_1 = I_s$;

2 Set $R_1 = \left(W_1^\top A\right)^\top$ and $S_1 = AV_1$;

3 For $j = 1, 2, \ldots, m$ Do:

- $A_j = W_j^\top S_j$
- $R_j = R_j - W_j A_j^\top$ *and* $S_j = S_j - V_j A_j$
- *Compute the QR decomposition:* $R_j = W_{j+1} B_{j+1}^\top$ *and* $S_j = V_{j+1} C_{j+1}$
- *Compute the singular value decomposition:* $W_{j+1}^\top V_{j+1} = \tilde{U}_j \sum_j \tilde{V}_j^\top$
- $B_{j+1} = B_{j+1} \tilde{U}_j \sum_j^{\frac{1}{2}}$ *and* $C_{j+1} = \sum_j^{\frac{1}{2}} \tilde{V}_j^\top C_{j+1}$
- $W_{j+1} = W_{j+1} \tilde{U}_j \sum_j^{-\frac{1}{2}}$ *and* $V_{j+1} = V_{j+1} \tilde{V}_j \sum_j^{-\frac{1}{2}}$
- $R_{j+1} = \left(W_{j+1}^\top A - C_{j+1} W_j^\top\right)^\top$ *and* $S_{j+1} = AV_{j+1} - V_j B_{j+1}$

End Do.

Notice that a breakdown may occur in Algorithm 2 if $R_j^\top S_j$, S_j or R_j is not of full rank. In [76], several strategies are proposed to overcome breakdowns and near-breakdowns in order to preserve the numerical stability of the block Lanczos process.

Denote $\mathcal{V}_m = [V_1, V_2, \ldots, V_m]$ and $\mathcal{W}_m = [W_1, W_2, \ldots, W_m]$ two $n \times ms$ matrices. This algorithm generates a nonsymmetric banded block tridiagonal matrix T_m, similar to A, expressed as:

$$T_m = \begin{pmatrix} A_1 & B_2 & & & \\ C_2 & A_2 & B_3 & & \\ & \ddots & \ddots & \ddots & \\ & & C_{m-1} & A_{m-1} & B_m \\ & & & C_m & A_m \end{pmatrix} \tag{5.1.4}$$

where A_i, B_i and C_i are the blocks given in Algorithm 2.

Proposition 5.1.1 *Assume that the Algorithm 2 does not break down before the $m-$th step. Then $\{V_1, V_2, \ldots, V_m\}$ and $\{W_1, W_2, \ldots, W_m\}$ form bi-orthogonal bases of the block Krylov subspaces $K_m(A, V_1)$ and $K_m(A^\top, W_1)$, respectively. Then, the following relations hold:*

$$A\mathcal{V}_m = \mathcal{V}_m T_m + \mathcal{V}_{m+1} C_{m+1} E_m^\top, \tag{5.1.5}$$

$$\mathcal{W}_m^\top A = T_m \mathcal{W}_m^\top + E_m B_{m+1} \mathcal{W}_{m+1}^\top, \tag{5.1.6}$$

$$\mathcal{W}_m^\top A \mathcal{V}_m = T_m. \tag{5.1.7}$$

where $E_m^\top = (0_s, \ldots, 0_s, I_s) \in R^{s \times ms}$.

5.1.2 Block nonsymmetric Lanczos method for linear systems

Consider the linear system (4.3.6) and let $X_0 \in \mathbb{R}^{n \times s}$ be an initial guess of the solution and $R_0 = B - AX_0$ the corresponding residual. Set $R_0 = V_1 \mathcal{R}$ (QR factorization) and let W_1 be a chosen matrix in $\mathbb{R}^{n \times s}$ satisfying $W_1^\top V_1 = I_s$. Assume that the matrices V_1 and W_1 are of full rank. The block nonsymmetric Lanczos method consists in finding a sequence of matrices X_m, defined by two conditions:

$$X_m - X_0 \in K_m(A, V_1) \tag{5.1.8}$$

$$R_m = B - AX_m \perp K_m(A^\top, W_1) \tag{5.1.9}$$

Using the first condition (5.1.8), the approximate solution can be expressed as:

$$X_m = X_0 + \mathcal{V}_m \mathcal{Y}_m, \tag{5.1.10}$$

where, due to the second condition (5.1.9), $\mathcal{Y}_m = [y_1, \ldots, y_m]$ satisfies the following relation:

$$\begin{aligned} \mathcal{W}_m^\top R_m &= \mathcal{W}_m^\top R_0 - \mathcal{W}_m^\top A \mathcal{V}_m \mathcal{Y}_m \\ &= 0_{ms \times s}. \end{aligned} \tag{5.1.11}$$

If $T_m = \mathcal{W}_m^\top A \mathcal{V}_m$ is nonsingular, it follows that:

$$\mathcal{Y}_m = T_m^{-1} \mathcal{W}_m^\top R_0. \tag{5.1.12}$$

Since $\mathcal{W}_m^\top R_0 = \mathcal{R} E_1$ with $E_1^\top = (I_s, 0_s, \ldots, 0_s) \in \mathbb{R}^{s \times ms}$, we obtain:

$$\mathcal{Y}_m = T_m^{-1} \mathcal{R} E_1, \tag{5.1.13}$$

and consequently:

$$X_m = X_0 + \mathcal{V}_m T_m^{-1} \mathcal{R} E_1. \tag{5.1.14}$$

Definition 5.1.2 *T_m is said to be a block strongly nonsingular matrix if $det(T_k) \neq 0$, for $k = 1, \ldots, m$.*

We note that the matrix X_m given by the expression (5.1.14) is a Schur complement and we have the following result:

Theorem 5.1.3 *Let X_m be the approximate solution given by the relation* (5.1.14)*. If the matrix T_m given by* (5.1.7) *is block strongly nonsingular, then X_m is a Schur complement and it may be computed as follows:*

$$X_m = X_{m-1} + G_m \left(T_m / T_{m-1}\right)^{-1} G'_m, \tag{5.1.15}$$

where G_m and G'_m are matrices given by:

$$G_m = \left(\begin{pmatrix} V_m & V_1 & \ldots & V_{m-1} \\ 0_s & & & \\ \vdots & & T_{m-1} & \\ 0_s & & & \\ B_m & & & \end{pmatrix} / T_{m-1} \right), \quad G'_m = \left(\begin{pmatrix} 0_s & 0_s & \ldots & 0_s & C_m \\ \mathcal{R} & & & & \\ 0_s & & T_{m-1} & & \\ \vdots & & & & \\ 0_s & & & & \end{pmatrix} / T_{m-1} \right). \tag{5.1.16}$$

Proof 1 *At step m, the approximate solution is expressed as a Schur complement which gives:*

$$X_m = X_0 + \mathcal{V}_m T_m^{-1} \mathcal{R} E_1 \tag{5.1.17}$$

$$= -\left(-X_0 - \mathcal{V}_m T_m^{-1} \mathcal{R} E_1\right) \tag{5.1.18}$$

$$= -\left(\begin{pmatrix} -X_0 & V_1 & \ldots & V_m \\ \mathcal{R} & & & \\ 0_s & & T_m & \\ \vdots & & & \\ 0_s & & & \end{pmatrix} / T_m \right) \tag{5.1.19}$$

$$= -\left(\begin{pmatrix} -X_0 & V_1 & \ldots & V_{m-1} & V_m \\ \mathcal{R} & & & & 0_s \\ 0_s & & T_{m-1} & & \vdots \\ \vdots & & & & B_m \\ 0_s & 0_s & \ldots & C_m & A_m \end{pmatrix} / T_m \right). \tag{5.1.20}$$

Using the matrix Sylvester identity (1.4.11)*, the matrix X_m can be calculated recursively as follows:*

$$X_m = X_{m-1} + (E_m/T_{m-1})\,(T_m/T_{m-1})^{-1}\,(E'_m/T_{m-1})\,, \tag{5.1.21}$$

where

$$E_m = \begin{pmatrix} V_1 & \dots & V_{m-1} & V_m \\ & & & 0_s \\ & T_{m-1} & & \vdots \\ & & & 0_s \\ & & & B_m \end{pmatrix} \quad \textit{and} \quad E'_m = \begin{pmatrix} \mathcal{R} & & & & \\ 0_s & & & & \\ \vdots & & T_{m-1} & & \\ 0_s & & & & \\ 0_s & 0_s & \dots & 0_s & C_m \end{pmatrix}. \tag{5.1.22}$$

Using the properties (1.4.7) *and* (1.4.8)*, we get:*

$$(E_m/T_{m-1}) = \left(\begin{pmatrix} V_1 & \dots & V_{m-1} & V_m \\ & & & 0_s \\ & T_{m-1} & & \vdots \\ & & & 0_s \\ & & & B_m \end{pmatrix} /T_{m-1} \right) = \left(\begin{pmatrix} V_m & V_1 & \dots & V_{m-1} \\ 0_s & & & \\ \vdots & & T_{m-1} & \\ 0_s & & & \\ B_m & & & \end{pmatrix} /T_{m-1} \right) = G_m, \tag{5.1.23}$$

$$(E'_m/T_{m-1}) = \left(\begin{pmatrix} \mathcal{R} & & & & \\ 0_s & & & & \\ \vdots & & T_{m-1} & & \\ 0_s & & & & \\ 0_s & 0_s & \dots & 0_s & C_m \end{pmatrix} /T_{m-1} \right) = \left(\begin{pmatrix} 0_s & 0_s & \dots & 0_s & C_m \\ \mathcal{R} & & & & \\ 0_s & & T_{m-1} & & \\ \vdots & & & & \\ 0_s & & & & \end{pmatrix} /T_{m-1} \right) = G'_m. \tag{5.1.24}$$

□

The matrices G_m and G'_m used to calculate the approximate solution X_m verify the following result:

Theorem 5.1.4 *Let G_m and G'_m be the matrices defined by* (5.1.16)*. If T_m is a block strongly nonsingular matrix, then:*

$$G_m = V_m - G_{m-1}(T_{m-1}/T_{m-2})^{-1}B_m, \tag{5.1.25}$$

$$G'_m = -C_m(T_{m-1}/T_{m-2})^{-1}G'_{m-1}. \tag{5.1.26}$$

Proof 2 *Note that the matrix G_m is a Schur complement given by:*

$$G_m = \left(\begin{pmatrix} V_m & V_1 & \dots & V_{m-1} \\ 0_s & & & \\ \vdots & & T_{m-1} & \\ 0_s & & & \\ B_m & & & \end{pmatrix} /T_{m-1} \right) = \left(\begin{pmatrix} V_m & V_1 & \dots & V_{m-2} & V_{m-1} \\ 0_s & & & & 0_s \\ \vdots & & T_{m-2} & & \vdots \\ 0_s & & & & B_{m-1} \\ B_m & 0_s & \dots & C_{m-1} & A_{m-1} \end{pmatrix} /T_{m-1} \right). \tag{5.1.27}$$

Applying the matrix Sylvester identity, we obtain:

$$
\begin{aligned}
G_m = & \left(\begin{pmatrix} V_m & V_1 \dots V_{m-2} \\ 0_s & \\ \vdots & T_{m-2} \\ 0_s & \end{pmatrix} / T_{m-2} \right) \\
& - \left(\begin{pmatrix} V_1 \dots V_{m-2} & V_{m-1} \\ & 0_s \\ T_{m-2} & \vdots \\ & B_{m-1} \end{pmatrix} / T_{m-2} \right) \left(T_{m-1}/T_{m-2} \right)^{-1} \left(\begin{pmatrix} 0_s & \\ \vdots & T_{m-2} \\ 0_s & \\ B_m & 0_s \dots C_{m-1} \end{pmatrix} / T_{m-2} \right) \qquad (5.1.28) \\
= & V_m - \left(\begin{pmatrix} V_1 \dots V_{m-2} & V_{m-1} \\ & 0_s \\ T_{m-2} & \vdots \\ & B_{m-1} \end{pmatrix} / T_{m-2} \right) \left(T_{m-1}/T_{m-2} \right)^{-1} B_m \qquad (5.1.29) \\
= & V_m - G_{m-1} \left(T_{m-1}/T_{m-2} \right)^{-1} B_m. \qquad (5.1.30)
\end{aligned}
$$

Similarly, we apply the matrix Sylvester identity to G'_m which is also a Schur complement expressed as:

$$
G'_m = \left(\begin{pmatrix} 0_s & 0_s & \dots & 0_s & C_m \\ \mathcal{R} & & & & \\ 0_s & & T_{m-1} & & \\ \vdots & & & & \\ 0_s & & & & \end{pmatrix} / T_{m-1} \right) = \left(\begin{pmatrix} 0_s & 0_s & \dots & 0_s & C_m \\ \mathcal{R} & & & & 0_s \\ 0_s & & T_{m-2} & & \vdots \\ \vdots & & & & B_{m-1} \\ 0_s & 0_s & \dots & C_{m-1} & A_{m-1} \end{pmatrix} / T_{m-1} \right). \qquad (5.1.31)
$$

Thus, we obtain:

$$
\begin{aligned}
G'_m = & \left(\begin{pmatrix} 0_s & 0_s \dots 0_s \\ \mathcal{R} & \\ \vdots & T_{m-2} \\ 0_s & \end{pmatrix} / T_{m-2} \right) \\
& - \left(\begin{pmatrix} 0_s \dots 0_s & C_m \\ & 0_s \\ T_{m-2} & \vdots \\ & B_{m-1} \end{pmatrix} / T_{m-2} \right) \left(T_{m-1}/T_{m-2} \right)^{-1} \left(\begin{pmatrix} \mathcal{R} & \\ \vdots & T_{m-2} \\ 0_s & \\ 0_s & 0_s \dots C_{m-1} \end{pmatrix} / T_{m-2} \right) \qquad (5.1.32) \\
= & -C_m \left(T_{m-1}/T_{m-2} \right)^{-1} \left(\begin{pmatrix} 0_s & 0_s \dots C_{m-1} \\ \mathcal{R} & \\ \vdots & T_{m-2} \\ 0_s & \end{pmatrix} / T_{m-2} \right) = -C_m (T_{m-1}/T_{m-2})^{-1} G'_{m-1}. \qquad (5.1.33)
\end{aligned}
$$

□

The following theorem gives a recursive expression to calculate the matrix (T_m/T_{m-1}). This result can be easily proved using the matrix Sylvester identity.

Theorem 5.1.5 *At step m, the matrix T_m is assumed to be block strongly nonsingular. Then the matrix (T_m/T_{m-1}) can be determined as follows:*

$$(T_m/T_{m-1}) = A_m - C_m \left(T_{m-1}/T_{m-2}\right)^{-1} B_m. \tag{5.1.34}$$

Proof 3 *The matrix (T_m/T_{m-1}) is given by:*

$$(T_m/T_{m-1}) = \left(\begin{pmatrix} A_1 & B_2 & 0_s & \dots & 0_s \\ C_2 & \ddots & \ddots & \ddots & \vdots \\ 0_s & \ddots & \ddots & \ddots & 0_s \\ \vdots & \ddots & \ddots & \ddots & B_m \\ 0_s & \dots & 0_s & C_m & A_m \end{pmatrix} /T_{m-1} \right) = \left(\begin{pmatrix} & & 0_s \\ & T_{m-1} & \vdots \\ & & 0_s \\ & & B_m \\ 0_s \dots 0_s & C_m & A_m \end{pmatrix} /T_{m-1} \right). \tag{5.1.35}$$

Using the relation (1.4.6)*, we can write:*

$$(T_m/T_{m-1}) = \left(\begin{pmatrix} A_m & 0_s \dots 0_s & C_m \\ 0_s & & \\ \vdots & T_{m-1} & \\ 0_s & & \\ B_m & & \end{pmatrix} /T_{m-1} \right) = \left(\begin{pmatrix} A_m & 0_s \dots 0_s & C_m \\ 0_s & & 0_s \\ \vdots & T_{m-2} & \vdots \\ 0_s & & B_{m-1} \\ B_m & 0_s \dots C_{m-1} & A_{m-1} \end{pmatrix} /T_{m-1} \right). \tag{5.1.36}$$

The matrix Sylvester identity yields:

$$\begin{aligned} (T_m/T_{m-1}) = & \left(\begin{pmatrix} A_m & 0_s \dots 0_s \\ 0_s & \\ \vdots & T_{m-2} \\ 0_s & \end{pmatrix} /T_{m-2} \right) \\ & - \left(\begin{pmatrix} 0_s \dots 0_s & C_m \\ & 0_s \\ T_{m-2} & \vdots \\ & B_{m-2} \end{pmatrix} /T_{m-2} \right) \left(T_{m-1}/T_{m-2}\right)^{-1} \left(\begin{pmatrix} 0_s & \\ \vdots & T_{m-2} \\ 0_s & \\ B_m & 0_s \dots B_{m-1} \end{pmatrix} /T_{m-2} \right) \end{aligned} \tag{5.1.37}$$

$$= A_m - C_m \left(T_{m-1}/T_{m-2}\right)^{-1} B_m. \tag{5.1.38}$$

□

Utilizing the block nonsymmetric Lanczos procedure and Theorem 5.1.3, Theorem 5.1.4 and Theorem 5.1.5, we obtain the new algorithm for implementing the block nonsymmetric Lanczos method. It is given as follows:

Algorithm 3 Block Nonsymmetric Lanczos Method

1 Choose $X_0 \in \mathbb{R}^{n\times s}$*, set* $R_0 = B - AX_0$ *and compute* $[V_1, \mathcal{R}] = QR(R_0)$

2 Choose $W_1 \in \mathbb{R}^{n\times s}$ *such that* $W_1^\top V_1 = I_s$

3 Set $G_1 = V_1$*,* $G_1' = \mathcal{R}$*,* $R_1 = (W_1^\top A)^\top$ *and* $S_1 = AV_1$

4 Set $C_1 = 0$*,* $B_1 = 0$ *and* $(T_0/T_{-1}) = 0$

5 For $j = 1, 2, \ldots$ *Do:*

- $A_j = W_j^\top S_j$
- $R_{j+1} = R_j - W_j A_j^\top$ *and* $S_{j+1} = S_j - V_j A_j$
- *Compute the QR decomposition:* $R_{j+1} = W_{j+1}B_{j+1}^\top$ *and* $S_{j+1} = V_{j+1}C_{j+1}$
- *Compute the singular value decomposition:* $W_{j+1}^\top V_{j+1} = \tilde{U}_j \sum_j \tilde{V}_j^\top$
- $B_{j+1} = B_{j+1}\tilde{U}_j \sum_j^{\frac{1}{2}}$ *and* $C_{j+1} = \sum_j^{\frac{1}{2}} \tilde{V}_j^\top C_{j+1}$
- $W_{j+1} = W_{j+1}\tilde{U}_j \sum_j^{-\frac{1}{2}}$ *and* $V_{j+1} = V_{j+1}\tilde{V}_j \sum_j^{-\frac{1}{2}}$
- $R_{j+1} = \left(W_{j+1}^\top A - C_{j+1}W_j^\top\right)^\top$ *and* $S_{j+1} = AV_{j+1} - V_j B_{j+1}$
- $(T_j/T_{j-1}) = A_j - C_j \left(T_{j-1}/T_{j-2}\right)^{-1} B_j$
- $X_j = X_{j-1} + G_j \left(T_j/T_{j-1}\right)^{-1} G_j'$
- $G_{j+1} = V_{j+1} - G_j \left(T_j/T_{j-1}\right)^{-1} B_{j+1}$
- $G_{j+1}' = -C_{j+1} \left(T_j/T_{j-1}\right)^{-1} G_j'$

End Do.

5.2 Numerical experiments

In this section, we present some numerical examples to illustrate the effectiveness of the proposed algorithm for large and sparse equations. The test matrices are taken from the Matrix Market [77]. We compared the block nonsymmetric Lanczos method (Bl-NLM) with the block biconjugate gradient stabilized method (Bl-BiCGStab). The value of s is taken to be 20. The tests were stopped

as soon as $\frac{\|R_m\|_2}{\|R_0\|_2} \leq \text{eps}$.

Matrix	n	Time(s)		Iterations		eps
		Bl-BiCGStab	Bl-NLM	Bl-BiCGStab	Bl-NLM	
lund_b	147	0.34	0.11	147	16	10^{-15}
cdde5	961	2.39	1.33	100	39	10^{-8}
tols4000	4000	42.64	48.05	201	201	10^{-6}
tols2000	2000	9.83	9.06	101	97	10^{-6}
sherman4	1104	0.56	0.28	57	35	10^{-14}
bcsstk09	1083	22.23	3.56	1083	90	10^{-14}
add32	4960	74.39	52.01	249	128	10^{-15}

Table 5.1: Matrices and numerical results of Bl-NLM and Bl-BiCGStab.

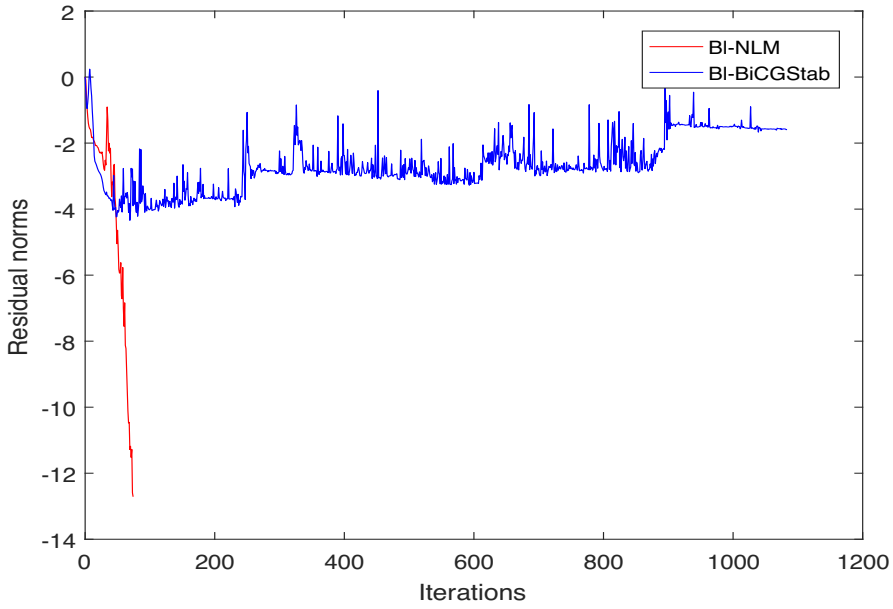

Figure 5.1: The convergence history of Bl-NLM and Bl-BiCGStab for bcsstk09.

Figure 5.1, Figure 5.2, Figure 5.3 and Figure 5.4 show the convergence rate of the methods Bl-NLM and Bl-BiCGStab for the matrices bcsstk09, lund_b, add32 and tols4000, respectively.

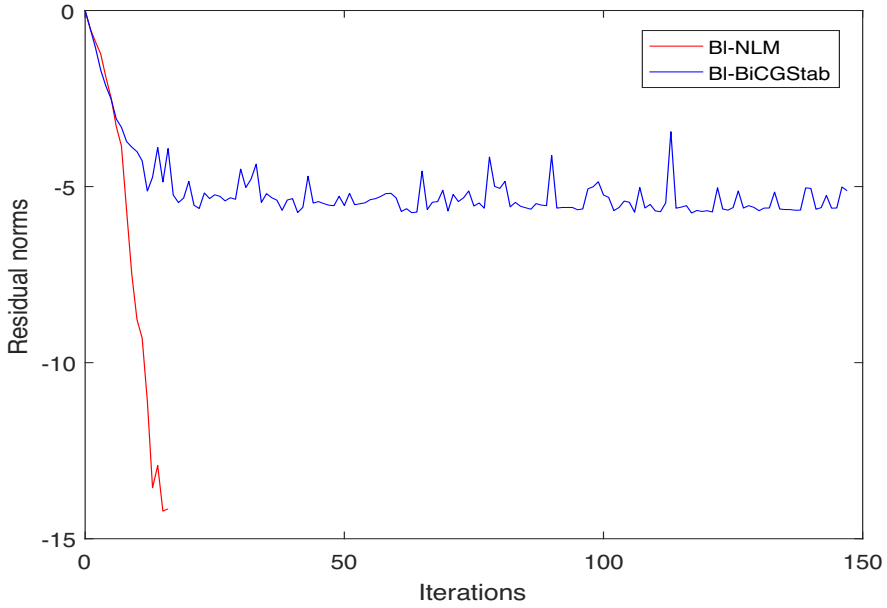

Figure 5.2: The convergence history of Bl-NLM and Bl-BiCGStab for lund_b.

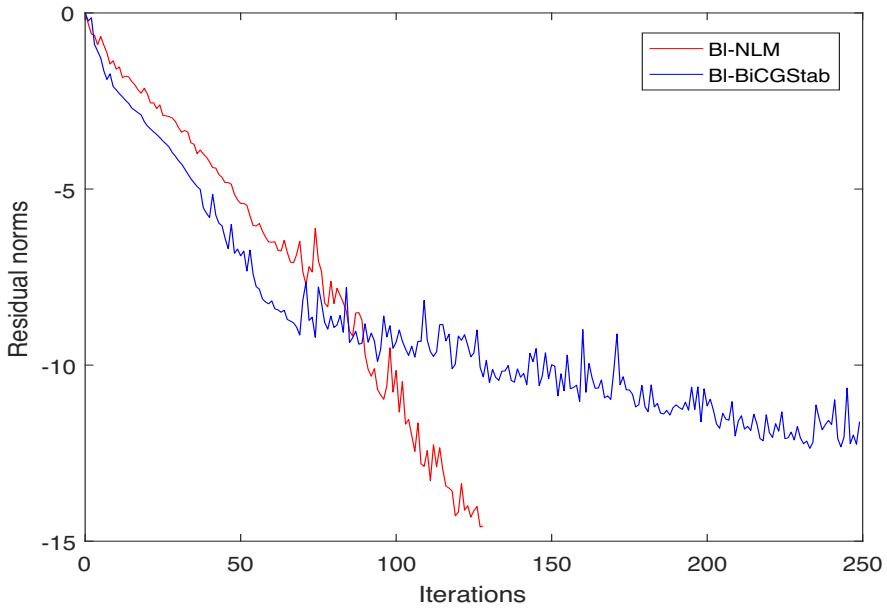

Figure 5.3: The convergence history of Bl-NLM and Bl-BiCGStab for add32.

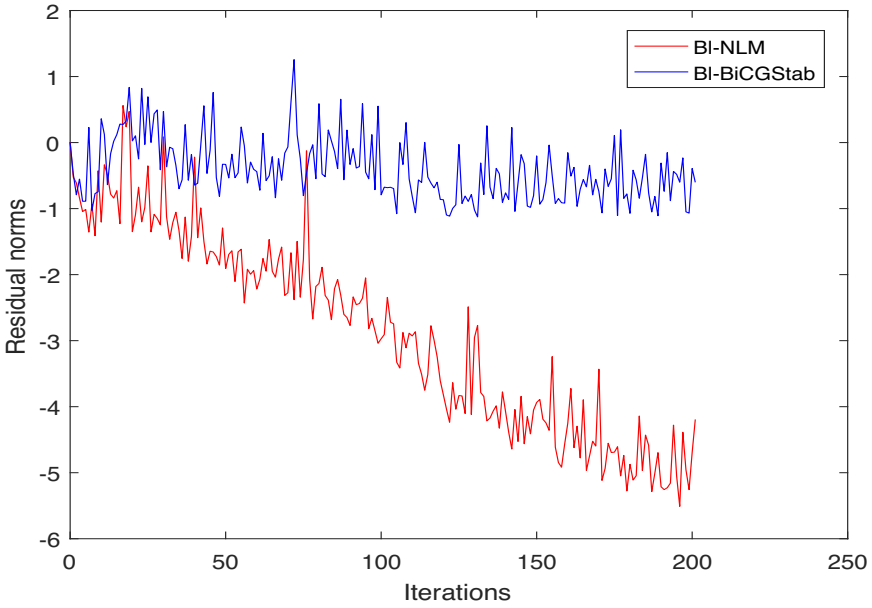

Figure 5.4: The convergence history of Bl-NLM and Bl-BiCGStab for tols4000.

For all these matrices, the Bl-NLM gives a better performance than the Bl-BiCGStab. In table 5.1, the computing time and the number of iterations were given. The BL-BiCGStab takes more time to give the results comparing to the Bl-NLM for the majority of the matrices listed in the table. Besides, the Bl-NLM converges in less number of iterations than the Bl-BiCGStab.

Conclusion

In the present chapter, we use the Schur complement and the block Lanczos process to define the block nonsymmetric Lanczos method for solving large linear systems with multiple right-hand sides. The numerical examples show that the proposed algorithm is effective and less expensive than other algorithms for some matrices.

Chapter 6

Direct global Lanczos method for large linear systems with multiple right-hand sides

In several practical situations, it is recommended to operate on a block of vectors instead of operating on a single vector, which is the case of solving large linear systems with the same coefficient matrix and various right-hand sides. The global Lanczos method is a projection method that has for purpose the resolution of all the linear systems simultaneously rather than using a standard iterative method for each system. The global Lanczos method has been frequently used for solving large linear problems. In 2005, Jbilou et al. [70] derived the global Lanczos process to construct biorthonormal bases and gave some of its properties. They introduced then some global Lanczos-based alorithms, such as the global BiCG algorithm and the global BiCGStab algorithm. In [72], Heyouni et al. considered matrix Krylov subspace methods based on the global Arnoldi process and on the global Lanczos process for solving large coupled Lyapunov matrix equations. The global Lanczos method was also applied in [78] for solving ill-posed linear matrix equations stemming from the restoration of blurred and noisy images. It was showed that restoring the available image can be accomplished using Tikhonov regularization based on the global Lanczos method and its connection to Gauss-type quadrature rules. In [79], Barkouki et al. proposed a rational version of the global Lanczos algorithm for model-order reduction problems using multi-point moment matching-based methods. In the present chapter, we are aimed at using the global nonsymmetric Lanczos algorithm with some Schur complements identities in order to obtain recursive formulas that help constructing a new variant of the global nonsymmetric Lanczos method.

6.1 Global nonsymmetric Lanczos method

The global Lanczos method is another extension of the classical Lanczos method as the block Lanczos method presented in the previous chapter. Although, this method differs from the standard and the block Lanczos methods in that it uses the Frobenius inner product between matrices, which allows using matrix-matrix products throughout the implementation. The global Lanczos method

is based on the global oblique projection of the initial residual onto a matrix Krylov subspace in order to reduce a large sparse matrix to a small tridiagonal one.

6.1.1 Global nonsymmetric Lanczos procedure

Given an $n \times n$ real nonsymmetric and nonsingular matrix A and initial $n \times s$ block vectors V_1 and W_1 such that $< W_1, V_1 >_F = 1$. The global Lanczos process consists in constructing two bi-orthogonal bases $\{V_1, V_2, \ldots, V_m\}$ and $\{W_1, W_2, \ldots, W_m\}$ of the matrix subspaces

$$K_m(A, V_1) = \text{span}\left\{V_1, AV_1, \ldots, A^{m-1}V_1\right\} \tag{6.1.1}$$

and

$$K_m(A^\top, W_1) = \text{span}\left\{W_1, A^\top W_1, \ldots, \left(A^\top\right)^{m-1} W_1\right\}, \tag{6.1.2}$$

respectively, such that:

$$< V_i, W_j >_F = \text{tr}\left(V_i^\top W_j\right) = \delta_{ij}, \tag{6.1.3}$$

for $i, j = 1, \ldots, m$. The algorithm given below executes m steps of the global Lanczos process applied to A:

Algorithm 4 Global Lanczos process *[70]*

1 Choose two matrices V_1 *and* $W_1 \in \mathbb{R}^{n \times s}$ *such that* $< W_1, V_1 >_F = 1$

2 Set $\beta_1 = \delta_1 = 0$ *and* $W_0 = V_0 = 0$

3 For $j = 1, 2, \ldots, m$ *Do:*

- $\alpha_j = tr(W_j^\top A V_j)$
- $\tilde{V}_{j+1} = AV_j - \alpha_j V_j - \beta_j V_{j-1}$
- $\tilde{W}_{j+1} = A^\top W_j - \alpha_j W_j - \delta_j W_{j-1}$
- $\delta_{j+1} = \left| tr\left(\tilde{V}_{j+1}^\top \tilde{W}_{j+1}\right)\right|^{1/2}$
- $\beta_{j+1} = tr\left(\tilde{V}_{j+1}^\top \tilde{W}_{j+1}\right) / \delta_{j+1}$
- $V_{j+1} = \tilde{V}_{j+1} / \delta_{j+1}$
- $W_{j+1} = \tilde{W}_{j+1} / \beta_{j+1}$

End Do.

This algorithm may break down in the j-th iteration if $\text{tr}\left(\tilde{V}_{j+1}^\top \tilde{W}_{j+1}\right) = 0$.

Denote $\mathcal{V}_m = [V_1, V_2, \ldots, V_m]$ and $\mathcal{W}_m = [W_1, W_2, \ldots, W_m]$ two $n \times ms$ matrices. Let T_m be the

$m \times m$ tridiagonal matrix defined by:

$$T_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & 0 \\ \delta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \delta_{m-1} & \alpha_{m-1} & \beta_m \\ 0 & & & \delta_m & \alpha_m \end{pmatrix} \tag{6.1.4}$$

where α_i, β_i and δ_i are the scalars defined in Algorithm 4.

Consider the matrices:

$$\tilde{T}_m^A = \begin{pmatrix} T_m \\ \delta_{m+1} e_m^\top \end{pmatrix} \in \mathbb{R}^{(m+1)\times m} \quad \text{and} \quad \tilde{T}_m^{A^\top} = \begin{pmatrix} T_m^\top \\ \beta_{m+1} e_m^\top \end{pmatrix} \in \mathbb{R}^{(m+1)\times m} \tag{6.1.5}$$

with $e_m = (0, \ldots, 0, 1)^\top \in \mathbb{R}^m$.

From now on, we will use the notation $*$ given in [71] to define the following product:

$$\mathcal{V}_m * y = \sum_{i=1}^{m} y^i V_i = \mathcal{V}_m (y \otimes Is), \tag{6.1.6}$$

where $y = (y^1, \ldots, y^m)^\top$ is a certain vector of $\mathbb{R}^m$. Similarly, we have:

$$\mathcal{V}_m * T_m = [\mathcal{V}_m * T_{.,1}, \ldots, \mathcal{V}_m * T_{.,m}], \tag{6.1.7}$$

where $T_{.,i}$ is the i-th column of the matrix T_m. Using these notations, we have the following result:

Proposition 6.1.1 *[70] Assume that the global Lanczos algorithm does not stop before the $m-$th step. Then the sets $\{V_1, V_2, \ldots, V_m\}$ and $\{W_1, W_2, \ldots, W_m\}$ form bi-orthogonal bases of the matrix Krylov subspaces $K_m(A, V_1)$ and $K_m(A^\top, W_1)$ respectively. Then, we have the following relations:*

$$A\mathcal{V}_m = \mathcal{V}_m * T_m + \delta_{m+1}[0_{n\times s}, \ldots, 0_{n\times s}, V_{m+1}], \tag{6.1.8}$$

$$A\mathcal{V}_m = \mathcal{V}_{m+1} * \tilde{T}_m^A, \tag{6.1.9}$$

$$A^\top \mathcal{W}_m = \mathcal{W}_m * T_m^\top + \beta_{m+1}[0_{n\times s}, \ldots, 0_{n\times s}, W_{m+1}], \tag{6.1.10}$$

$$A^\top \mathcal{W}_m = \mathcal{W}_{m+1} * \tilde{T}_m^{A^\top}, \tag{6.1.11}$$

$$\mathcal{W}_m^\top \diamond A\mathcal{V}_m = T_m. \tag{6.1.12}$$

6.1.2 Global nonsymmetric Lanczos method for linear systems

Consider the linear system (4.3.6). Let X_0 be an initial guess of the solution and $R_0 = B - AX_0$ be the corresponding residual. The global Lanczos method for the resolution of the system (4.3.6) consists in determining a sequence of approximations X_m such that:

$$X_m - X_0 \in K_m(A, R_0), \tag{6.1.13}$$

$$R_m = B - AX_m \perp_F K_m(A^\top, \tilde{R}_0), \tag{6.1.14}$$

where $\tilde{R}_0$ is a $n \times s$ matrix chosen so that $< R_0, \tilde{R}_0 >_F \neq 0$.

Let $\{V_1, V_2, \ldots, V_m\}$ and $\{W_1, W_2, \ldots, W_m\}$ be the sets of matrices defined by Algorithm 4 and which span the matrix Krylov subspaces $K_m(A, R_0)$ and $K_m(A^\top, \tilde{R}_0)$, respectively, with the initialization $V_1 = \dfrac{R_0}{\|R_0\|_F}$ and W_1 satisfies $< V_1, W_1 >_F = 1$.

The relation (6.1.13) yields:

$$X_m = X_0 + \mathcal{V}_m * y_m, \tag{6.1.15}$$

where y_m is the vector of $\mathbb{R}^m$ obtained by the relation (6.1.14):

$$< R_0 - A\mathcal{V}_m * y_m, W_i >_F = 0, \tag{6.1.16}$$

which is equivalent to:

$$< R_0, W_i >_F = < A\mathcal{V}_m * y_m, W_i >_F, \tag{6.1.17}$$

for $i = 1, \ldots, m$. This relation can also be written in the form:

$$\sum_{j=1}^{m} y_m^j \mathrm{tr}\left(W_1^\top A V_j\right) = \|R_0\|_F \quad \text{and} \quad \sum_{j=1}^{m} y_m^j \mathrm{tr}\left(W_i^\top A V_j\right) = 0, \tag{6.1.18}$$

for $i = 2, \ldots, m$. Then, we get the following linear system:

$$T_m y_m = \|R_0\|_F \, e_1^{(m)}, \tag{6.1.19}$$

where $e_1^{(m)} = (1, 0, \ldots, 0)^\top \in \mathbb{R}^m$. If the matrix T_m is nonsingular, then the approximations X_m obtained by the global Lanczos method, are given by the following expression:

$$X_m = X_0 + \|R_0\|_F \, \mathcal{V}_m * T_m^{-1} e_1^{(m)}. \tag{6.1.20}$$

We note that the matrix X_m given by (6.1.20) is a Schur complement and we have the following result:

Theorem 6.1.2 *Let X_m be the approximate matrix of X given by the relation* (6.1.20)*. If the matrix T_m is block strongly nonsingular, then X_m is a Schur complement and it can be expressed in the following form:*

$$X_m = X_{m-1} + G_m \left(T_m \otimes Is / T_{m-1} \otimes Is\right)^{-1} G_m', \tag{6.1.21}$$

where the matrices G_m and G_m' are defined by:

$$G_m = \left(\begin{pmatrix} V_m & V_1 & \ldots & V_{m-1} \\ 0_s & & & \\ \vdots & & T_{m-1} \otimes Is & \\ 0_s & & & \\ \beta_m I_s & & & \end{pmatrix} / T_{m-1} \otimes Is \right), \quad G_m' = \left(\begin{pmatrix} 0_s & 0_s & \ldots & \delta_m I_s \\ \|R_0\|_F \, I_s & & & \\ 0_s & & T_{m-1} \otimes Is & \\ \vdots & & & \\ 0_s & & & \end{pmatrix} / T_{m-1} \otimes Is \right). \tag{6.1.22}$$

Proof 4 *According to* (6.1.20)*, we have:*

$$X_m = X_0 + \|R_0\|_F \mathcal{V}_m * \left(T_m^{-1} e_1^{(m)}\right). \tag{6.1.23}$$

Thus:

$$\begin{aligned}
X_m &= X_0 + \|R_0\|_F \mathcal{V}_m \left(T_m^{-1} e_1^{(m)} \otimes I_s\right) && (6.1.24)\\
&= X_0 + \|R_0\|_F \mathcal{V}_m \left(T_m \otimes I_s\right)^{-1} \left(e_1^{(m)} \otimes I_s\right) && (6.1.25)\\
&= -\left(-X_0 - \mathcal{V}_m \left(T_m \otimes I_s\right)^{-1} \left(\|R_0\|_F e_1^{(m)} \otimes I_s\right)\right) && (6.1.26)\\
&= -\left(\begin{pmatrix} -X_0 & V_1 & \dots & V_m \\ \|R_0\|_F I_s & & & \\ 0_s & & T_m \otimes Is & \\ \vdots & & & \\ 0_s & & & \end{pmatrix} / T_m \otimes Is\right) && (6.1.27)\\
&= -\left(\begin{pmatrix} -X_0 & V_1 & \dots & V_{m-1} & V_m \\ \|R_0\|_F I_s & & & & 0_s \\ 0_s & & T_{m-1} \otimes Is & & \vdots \\ \vdots & & & & \beta_m I_s \\ 0_s & 0_s & \dots & \delta_m I_s & \alpha_m I_s \end{pmatrix} / T_m \otimes Is\right). && (6.1.28)
\end{aligned}$$

Using the matrix Sylvester identity (1.4.11)*, the matrix X_m can be expressed as follows:*

$$X_m = X_{m-1} + \left(E_m / T_{m-1} \otimes Is\right) \left(T_m \otimes Is / T_{m-1} \otimes Is\right)^{-1} \left(E'_m / T_{m-1} \otimes Is\right), \tag{6.1.29}$$

where

$$E_m = \begin{pmatrix} V_1 & \dots & V_{m-1} & V_m \\ & & & 0_s \\ & T_{m-1} \otimes Is & & \vdots \\ & & & 0_s \\ & & & \beta_m I_s \end{pmatrix} \quad \textit{and} \quad E'_m = \begin{pmatrix} \|R_0\|_F I_s & & & \\ 0_s & & & \\ \vdots & & T_{m-1} \otimes Is & \\ 0_s & & & \\ 0_s & 0_s & \dots & \delta_m I_s \end{pmatrix}. \tag{6.1.30}$$

According to the properties (1.4.7) *and* (1.4.8)*, we have:*

$$\begin{aligned}
\left(E_m / T_{m-1} \otimes Is\right) &= \left(\begin{pmatrix} V_1 & \dots & V_{m-1} & V_m \\ & & & 0_s \\ T_{m-1} \otimes Is & & & \vdots \\ & & & 0_s \\ & & & \beta_m I_s \end{pmatrix} / T_{m-1} \otimes Is\right) \\
&= \left(\begin{pmatrix} V_m & V_1 & \dots & V_{m-1} \\ 0_s & & & \\ \vdots & & T_{m-1} \otimes Is & \\ 0_s & & & \\ \beta_m I_s & & & \end{pmatrix} / T_{m-1} \otimes Is\right) = G_m, && (6.1.31)
\end{aligned}$$

$$(E'_m/T_{m-1}\otimes Is) = \left(\begin{pmatrix}\begin{pmatrix}\|R_0\|_F I_s \\ 0_s \\ \vdots \\ 0_s \\ 0_s\end{pmatrix} & \begin{matrix} \\ \\ T_{m-1}\otimes Is \\ \\ 0_s \ \ \ldots \ \ \delta_m I_s\end{matrix}\end{pmatrix} /T_{m-1}\otimes Is\right)$$

$$= \left(\begin{pmatrix}\begin{pmatrix}0_s \\ \|R_0\|_F I_s \\ 0_s \\ \vdots \\ 0_s\end{pmatrix} & \begin{matrix}0_s \ \ \ldots \ \ \delta_m I_s \\ \\ T_{m-1}\otimes Is \\ \\ \end{matrix}\end{pmatrix} /T_{m-1}\otimes Is\right) = G'_m. \tag{6.1.32}$$

□

The matrices G_m and G'_m used to express the approximate solution X_m satisfy the following result.

Theorem 6.1.3 *Let G_m and G'_m be the matrices defined by* (6.1.22)*. If T_m is a block strongly nonsingular matrix, then:*

$$G_m = V_m - \beta_m G_{m-1}\left(T_{m-1}\otimes Is/T_{m-2}\otimes Is\right)^{-1}, \tag{6.1.33}$$

$$G'_m = -\delta_m\left(T_{m-1}\otimes Is/T_{m-2}\otimes Is\right)^{-1} G'_{m-1}. \tag{6.1.34}$$

Proof 5 *We can see that the matrices G_m and G'_m are Schur complements and may be expressed as:*

$$G_m = \left(\begin{pmatrix}\begin{pmatrix}V_m & V_1 & \ldots & V_{m-1} \\ 0_s & & & \\ \vdots & & T_{m-1}\otimes Is & \\ 0_s & & & \\ \beta_m I_s & & & \end{pmatrix} /T_{m-1}\otimes Is\end{pmatrix}\right), \tag{6.1.35}$$

$$= \left(\begin{pmatrix}\begin{pmatrix}V_m & V_1 & \ldots & V_{m-2} & V_{m-1} \\ 0_s & & & & 0_s \\ \vdots & & T_{m-2}\otimes Is & & \vdots \\ 0_s & & & & \beta_{m-1} I_s \\ \beta_m I_s & 0_s & \ldots & \delta_{m-1} I_s & \alpha_{m-1} I_s\end{pmatrix} /T_{m-1}\otimes Is\end{pmatrix}\right), \tag{6.1.36}$$

and

$$G'_m = \left(\begin{pmatrix} 0_s & 0_s & \dots & \delta_m I_s \\ \|R_0\|_F I_s & & & \\ 0_s & & T_{m-1} \otimes Is & \\ \vdots & & & \\ 0_s & & & \end{pmatrix} /T_{m-1} \otimes Is \right), \tag{6.1.37}$$

$$= \left(\begin{pmatrix} 0_s & 0_s & \dots & 0_s & \delta_m I_s \\ \|R_0\|_F I_s & & & & 0_s \\ \vdots & & T_{m-2} \otimes Is & & \vdots \\ 0_s & & & & \beta_{m-1} I_s \\ 0_s & 0_s & \dots & \delta_{m-1} I_s & \alpha_{m-1} I_s \end{pmatrix} /T_{m-1} \otimes Is \right). \tag{6.1.38}$$

Applying the matrix Sylvester identity, we get:

$$G_m = V_m - \beta_m \left(\begin{pmatrix} V_1 & \dots & V_{m-2} & V_{m-1} \\ & & & 0_s \\ & T_{m-1} \otimes Is & & \vdots \\ & & & 0_s \\ & & & \beta_{m-1} I_s \end{pmatrix} /T_{m-2} \otimes Is \right) \left(T_{m-1} \otimes Is / T_{m-2} \otimes Is \right)^{-1}, \tag{6.1.39}$$

$$= V_m - \beta_m G_{m-1} \left(T_{m-1} \otimes Is / T_{m-2} \otimes Is \right)^{-1}, \tag{6.1.40}$$

and

$$G'_m = -\delta_m \left(T_{m-1} \otimes Is / T_{m-2} \otimes Is \right)^{-1} \left(\begin{pmatrix} 0_s & 0_s & \dots & \delta_{m-1} I_s \\ \|R_0\|_F I_s & & & \\ 0_s & & T_{m-2} \otimes Is & \\ \vdots & & & \\ 0_s & & & \end{pmatrix} /T_{m-2} \otimes Is \right), \tag{6.1.41}$$

$$= -\delta_m \left(T_{m-1} \otimes Is / T_{m-2} \otimes Is \right)^{-1} G'_{m-1}. \tag{6.1.42}$$

□

The following theorem is an immediate consequence of the matrix Sylvester identity.

Theorem 6.1.4 *At step m, we assume that the matrix T_m is block strongly nonsingular, then we can calculate the matrix $((T_m \otimes Is)/(T_{m-1} \otimes Is))$ using the following expression:*

$$\left(T_m \otimes Is / T_{m-1} \otimes Is \right) = \alpha_m I_s - \delta_m \beta_m \left(T_{m-1} \otimes Is / T_{m-2} \otimes Is \right)^{-1}. \tag{6.1.43}$$

Proof 6 *The property* (1.4.6) *yields:*

$$(T_m \otimes Is / T_{m-1} \otimes Is) = \left(\begin{pmatrix} & & & 0_s \\ & T_{m-1} \otimes Is & & \vdots \\ & & & 0_s \\ & & & \beta_m I_s \\ 0_s & \dots & \delta_m I_s & \alpha_m I_s \end{pmatrix} / T_{m-1} \otimes Is \right), \tag{6.1.44}$$

$$= \left(\begin{pmatrix} \alpha_m I_s & 0_s & \dots & \delta_m I_s \\ 0_s & & & \\ \vdots & & T_{m-1} \otimes Is & \\ 0_s & & & \\ \beta_m I_s & & & \end{pmatrix} / T_{m-1} \otimes Is \right), \tag{6.1.45}$$

$$= \left(\begin{pmatrix} \alpha_m I_s & 0_s & \dots & 0_s & \delta_m I_s \\ 0_s & & & & 0_s \\ \vdots & & T_{m-2} \otimes Is & & \vdots \\ 0_s & & & & \beta_{m-1} I_s \\ \beta_m I_s & 0_s & \dots & \delta_{m-1} I_s & \alpha_{m-1} I_s \end{pmatrix} / T_{m-1} \otimes Is \right). \tag{6.1.46}$$

If we apply the matrix Sylvester identity, we immediately get the result :

$$(T_m \otimes Is / T_{m-1} \otimes Is) = \alpha_m I_s - \delta_m \beta_m \left(T_{m-1} \otimes Is / T_{m-2} \otimes Is\right)^{-1}. \tag{6.1.47}$$

□

Using the global Lanczos process and the three theorems above, we procure the new algorithm for implementing the global nonsymmetric Lanczos method. The algorithm is given as follows:

Algorithm 5 Direct Global Nonsymmetric Lanczos method

1 Choose an initial guess X_0 *and set* $R_0 = B - AX_0$

2 Set $\gamma = \|R_0\|_F$ *and* $V_1 = R_0/\gamma$

3 Choose $W_1 \in \mathbb{R}^{n \times s}$ *such that* $< W_1, V_1 >_F = 1$

4 Set $G_1 = V_1$, $G_1' = \gamma I_s$, $\beta_1 = \delta_1 = 0$, $W_0 = V_0 = 0$ *and* $(T_0 \otimes I_s / T_{-1} \otimes I_s) = 0$

5 For $j = 1, 2, \dots$ *Do:*

- $\alpha_j = tr(W_j^\top A V_j)$
- $\tilde{V}_{j+1} = AV_j - \alpha_j V_j - \beta_j V_{j-1}$
- $\tilde{W}_{j+1} = A^\top W_j - \alpha_j W_j - \delta_j W_{j-1}$
- $\delta_{j+1} = |tr(\tilde{V}_{j+1}^\top \tilde{W}_{j+1})|^{1/2}$
- $\beta_{j+1} = tr(\tilde{V}_{j+1}^\top \tilde{W}_{j+1})/\delta_{j+1}$

- $V_{j+1} = \tilde{V}_{j+1}/\delta_{j+1}$
- $W_{j+1} = \tilde{W}_{j+1}/\beta_{j+1}$
- $(T_j \otimes Is/T_{j-1} \otimes Is) = \alpha_j I_s - \delta_j\beta_j \left(T_{j-1} \otimes Is/T_{j-2} \otimes Is\right)^{-1}$
- $X_j = X_{j-1} + G_j \left(T_j \otimes Is/T_{j-1} \otimes Is\right)^{-1} G'_j$
- $G_{j+1} = V_{j+1} - \beta_{j+1} G_j \left(T_j \otimes Is/T_{j-1} \otimes Is\right)^{-1}$
- $G'_{j+1} = -\delta_{j+1} \left(T_j \otimes Is/T_{j-1} \otimes Is\right)^{-1} G'_j$

End Do.

6.2 Numerical examples

The current section describes some numerical experiments to examine the efficiency of the Global Nonsymmetric Lanczos Method (Gl-NLM) when applied to large linear systems with multiple right-hand sides. In these experiments, we compare the performance of the Gl-NLM with the Gl-BiCG, the Gl-BiCGStab and the restarted Gl-GMRES. Here, the Gl-GMRES is restarted every 20 iterations. The test matrices are taken from the Matrix Market [77]. The parameter s is taken as $s = 20$ for all the experiments. The stopping criterion in all iterative methods is $\frac{\|R_m\|_F}{\|R_0\|_F} \leq \text{eps}$.

Matrix	eps	CPU time(s)		Iterations		Storage(GB)	
		Gl-BiCGStab	Gl-NLM	Gl-BiCGStab	Gl-NLM	Gl-BiCGStab	Gl-NLM
662_bus	10^{-4}	1.06	2.51	130	130	1.47	1.47
685_bus	10^{-11}	4.73	9.46	685	685	1.39	1.39
e05r0300	10^{-1}	0.046	0.023	13	13	1.36	1.34
qh1484	10^{-3}	1.05	1.30	76	33	1.35	1.35
rdb2048l	10^{-3}	1.18	1.40	104	98	1.35	1.35
tols4000	10^{-3}	1.59	1.79	201	119	1.35	1.35

Table 6.1: Matrices and numerical results of Gl-NLM and Gl-BiCGStab.

Figure 6.1 and Figure 6.2 show the convergence rate of the methods Gl-NLM and Gl-BiCGStab for the matrices 662_bus and 685_bus, respectively. For both matrices, the Gl-NLM gives a better performance than the Gl-BiCGStab. In Table 6.1, the CPU time and the memory storage were given. The Gl-NLM takes more time to give the results comparing to the GL-BiCGStab but it converges in less number of iterations for the last three matrices.

As shown in Figure 6.3 and Figure 6.4, the convergence curves are very close, but the Gl-NLM is faster than the Gl-GMRES(20). For the matrix odep400b, the latter method converges in 42 iterations whereas the Gl-NLM method converges in about 21 iterations. Table 6.2 demonstrates that the Gl-NLM works better than the Gl-GMRES(20) for the different matrices listed above, in

terms of CPU time and storage required.

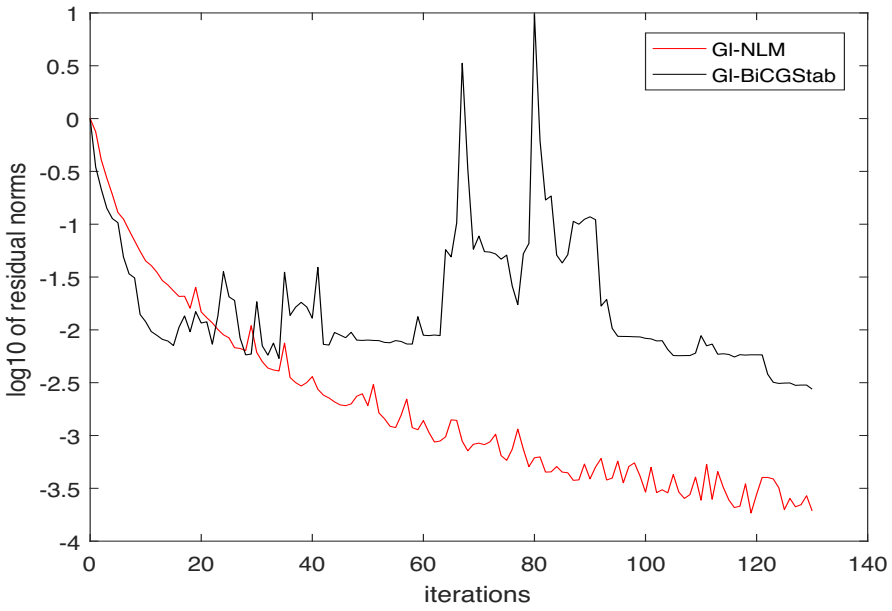

Figure 6.1: The convergence history of Gl-NLM and Gl-BiCGStab for 662_bus.

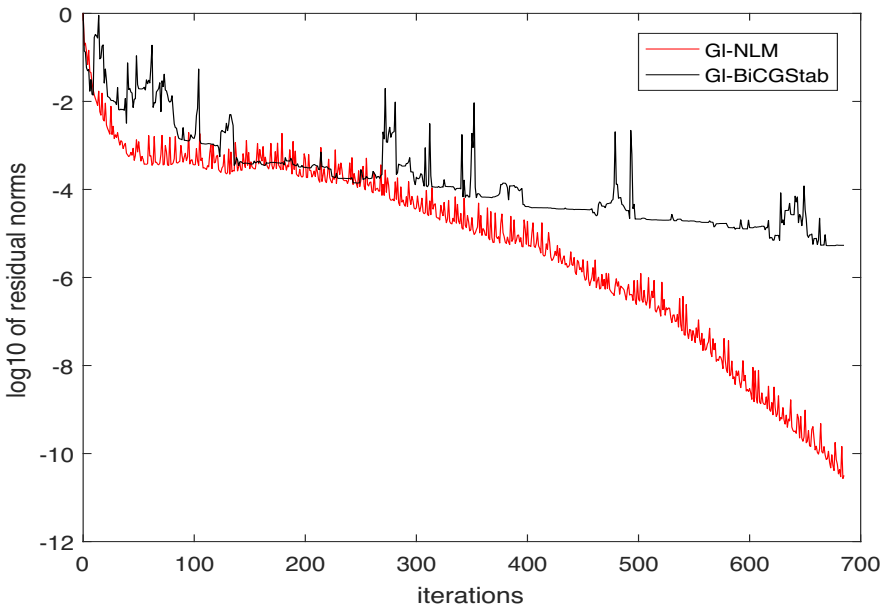

Figure 6.2: The convergence history of Gl-NLM and Gl-BiCGStab for 685_bus.

Matrix	eps	CPU time(s)		Iterations		Storage(GB)	
		Gl-GMRES	Gl-NLM	Gl-GMRES	Gl-NLM	Gl-GMRES	Gl-NLM
add32	10^{-15}	140.84	71.71	200	200	2.22	1.66
bfw398b	10^{-4}	0.18	0.10	21	20	1.36	1.35
bfw782b	10^{-9}	1.25	0.24	41	39	1.37	1.35
dwb512	10^{-14}	0.35	0.16	27	26	1.36	1.34
fidap007	10^{-5}	4.23	1.42	83	76	1.40	1.32
odep400b	10^{-16}	0.64	0.15	42	21	1.50	1.49

Table 6.2: Matrices and numerical results of Gl-NLM and Gl-GMRES(20).

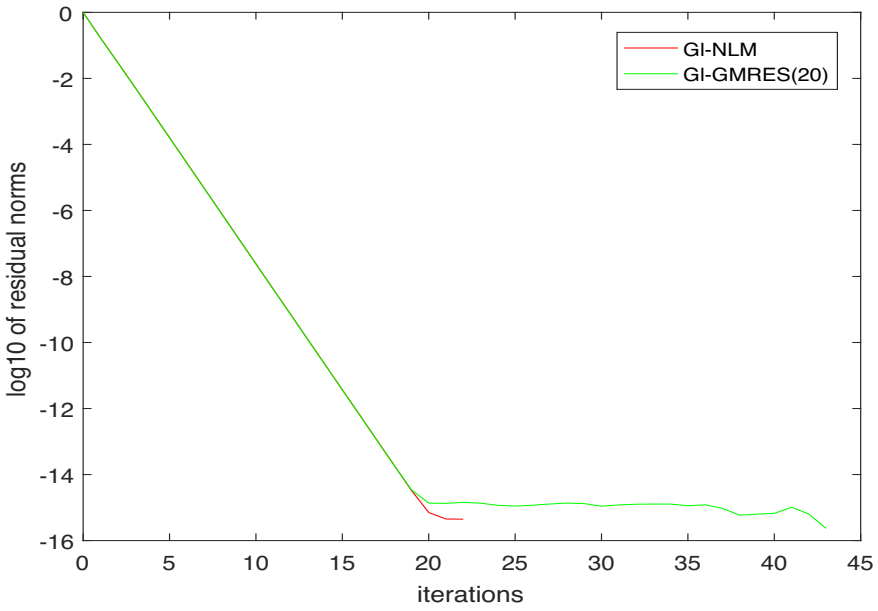

Figure 6.3: The convergence history of Gl-NLM and Gl-GMRES(20) for add32.

Matrix	eps	CPU time(s)		Iterations		Storage(GB)	
		Gl-BiCG	Gl-NLM	Gl-BiCG	Gl-NLM	Gl-BiCG	Gl-NLM
662_bus	10^{-3}	0.09	0.16	35	35	1.32	1.31
685_bus	10^{-3}	0.08	0.15	36	34	1.34	1.34
cry2500	10^{-2}	0.83	1.53	126	126	1.35	1.34
dwg961b	10^{-1}	0.53	1.13	50	50	1.34	1.34
odep400b	10^{-16}	0.02	0.05	20	21	1.35	1.35
bfw398b	10^{-5}	0.03	0.08	21	21	1.35	1.35

Table 6.3: Matrices and numerical results of Gl-NLM and Gl-BiCG.

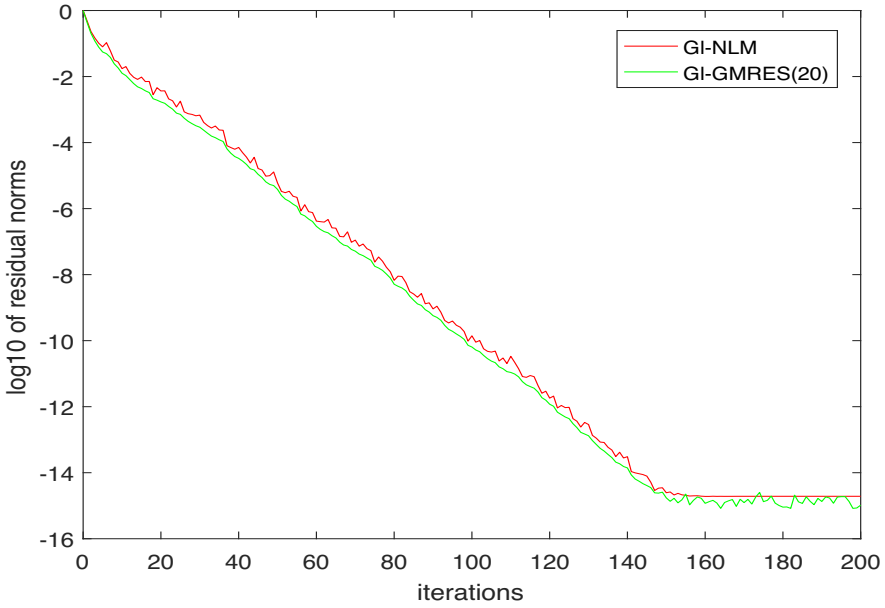

Figure 6.4: The convergence history of Gl-NLM and Gl-GMRES(20) for odep400b.

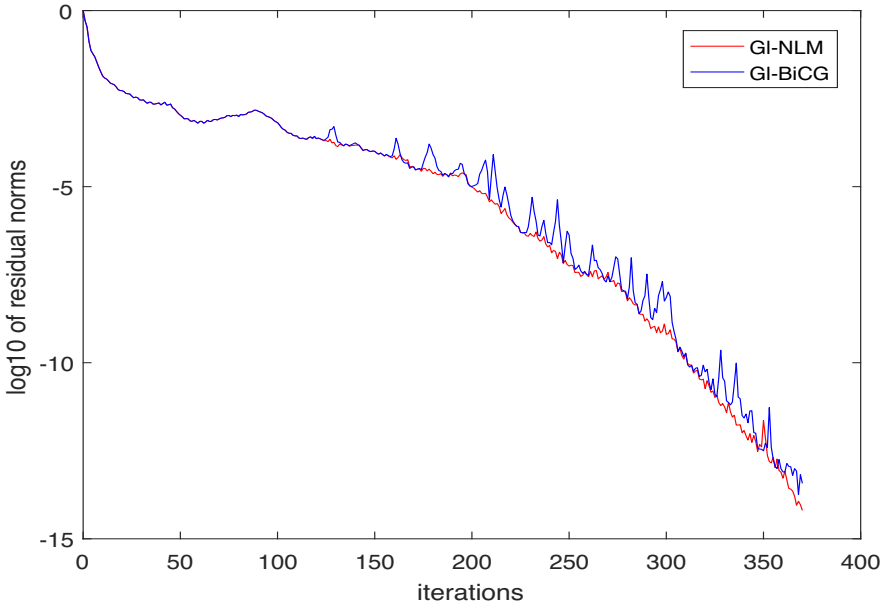

Figure 6.5: The convergence history of Gl-NLM and Gl-BiCG for bcsstk09.

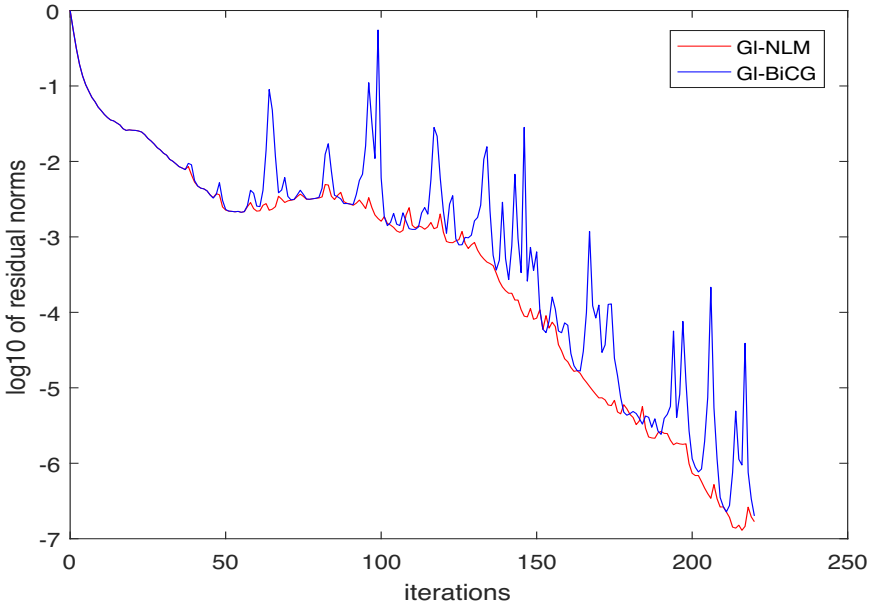

Figure 6.6: The convergence history of Gl-NLM and Gl-BiCG for fidap003.

Figure 6.5 and Figure 6.6 plot the curves where the convergence rate of the Gl-NLM and the Gl-BiCG methods is almost the same. According to Table 6.3, the Gl-NLM requires almost the same memory storage and number of iterations as the Gl-BiCG for different matrices. These results explain the fact that the two methods are similar, since the Gl-BiCG method is also derived from the nonsymmetric Lanczos process.

Conclusion

In this chapter, we derive a new method for solving large linear systems with multiple right-hand sides, basing on the global Lanczos process and some properties of the Schur complement. The numerical experiments show that the proposed approach is efficient than the other methods for some matrices.

Chapter 7

Global Lanczos method for the approximation of the transfer function

The mathematical description of a linear dynamical system is expressed as first-order differential equations known as the state equations given in the following state-space form :

$$\mathcal{S} : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \tag{7.0.1}$$

where the matrices $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times q}$ describe the properties of the system and are determined by the system structure and elements, $C^\top \in \mathbb{R}^{n\times p}$ is the output equation matrix, $x(t) \in \mathbb{R}^n$ is the internal variable, $u(t) \in \mathbb{R}^q$ and $y(t) \in \mathbb{R}^p$ are, respectively, the inputs and outputs of the system.

If $q = 1$ and $p = 1$, the dynamical system is called single input single output system (SISO); if $q > 1$ and $p > 1$, it is called multi-input multi-output system (MIMO); $q = 1$ and $p > 1$, it is called single input multi-output system (SIMO); $q > 1$ and $p = 1$, it is called multi-input single output system (MISO).

Using the Laplace transform, the state-space form of the state equations can be transformed into the transfer function $H : \mathbb{C} \to \mathbb{C}^{p\times q}$ defined as:

$$H(s) = C(sI - A)^{-1}B. \tag{7.0.2}$$

It represents the relation between the inputs and outputs of a linear dynamical system.

Numerous physical phenomena, such as wave propagation and heat transfer, are modeled accurately using linear dynamical systems. In the simulation and control of such systems, the dimension of the latter is quite large which causes limited computational resources and a poor accuracy due to the ill-conditioning of the system matrices. In order to surmount this difficulty, the model order reduction attempts to generate a smaller system whose characteristics are similar to those of the original system and which requires less computing time and cost. Many model reduction methods [80, 81, 82] have been used for this purpose. Examples include the Krylov based model reduction methods [83] which are based on matching the so-called moments of the original transfer function

around selected frequencies. Krylov based methods are iterative methods that are numerically efficient when applied to systems of high order such as those arising from structure dynamics, control systems, circuit simulations, computational electromagnetics and microelectromechanical systems [80, 84].

The goal of the present chapter is to introduce a new algorithm to approximate the transfer function associated to the state equations. The global Lanczos procedure is used as a model order reduction method. The proposed approach is based on the Schur complements properties which give recursive relations that are useful for the implementation.

7.1 Global nonsymmetric Lanczos process

Given an n by n real nonsymmetric and nonsingular matrix A and initial n by q block vectors V_1 and W_1 such that $< W_1, V_1 >_F = 1$. The global Lanczos process consists in constructing two bi-orthogonal bases $\{V_1, V_2, \ldots, V_m\}$ and $\{W_1, W_2, \ldots, W_m\}$ of the matrix Krylov subspaces:

$$K_m(A, V_1) = span\{V_1, AV_1, A^2V_1, \ldots, A^{m-1}V_1\} \tag{7.1.1}$$

and

$$K_m(A^\top, W_1) = span\{W_1, A^\top W_1, (A^\top)^2 W_1, \ldots, (A^\top)^{m-1} W_1\}, \tag{7.1.2}$$

respectively, such that

$$< V_i, W_j >_F = tr(V_i^\top W_j) = \delta_{ij}, \quad i, j = 1, \ldots, m. \tag{7.1.3}$$

The algorithm is given as follows:

Algorithm 6 Global Lanczos process

1 Choose two matrices V_1 *and* $W_1 \in \mathbb{R}^{n\times s}$ *such that* $< W_1, V_1 >_F = 1$

2 Set $\beta_1 = \delta_1 = 0$ *and* $W_0 = V_0 = 0$

3 For $j = 1, 2, \ldots, m$ *Do:*

- $\alpha_j = tr(W_j^\top A V_j)$
- $\tilde{V}_{j+1} = AV_j - \alpha_j V_j - \beta_j V_{j-1}$
- $\tilde{W}_{j+1} = A^\top W_j - \alpha_j W_j - \delta_j W_{j-1}$
- $\delta_{j+1} = \left| tr\left(\tilde{V}_{j+1}^\top \tilde{W}_{j+1}\right)\right|^{1/2}$
- $\beta_{j+1} = tr\left(\tilde{V}_{j+1}^\top \tilde{W}_{j+1}\right) / \delta_{j+1}$
- $V_{j+1} = \tilde{V}_{j+1} / \delta_{j+1}$
- $W_{j+1} = \tilde{W}_{j+1} / \beta_{j+1}$

End Do.

Denote $\mathcal{V}_m = [V_1, V_2, \ldots, V_m]$ and $\mathcal{W}_m = [W_1, W_2, \ldots, W_m]$ two $n \times mq$ matrices. The global Lanczos process generates an $m \times m$ tridiagonal matrix T_m similar to A, given by:

$$T_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & 0 \\ \delta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \delta_{m-1} & \alpha_{m-1} & \beta_m \\ 0 & & & \delta_m & \alpha_m \end{pmatrix} \tag{7.1.4}$$

where α_i, β_i and δ_i are the scalars defined in Algorithm 6.

Assume that the global Lanczos algorithm does not stop before the $m-$th step. Then the sets $\{V_1, V_2, \ldots, V_m\}$ and $\{W_1, W_2, \ldots, W_m\}$ form bi-orthogonal bases of the matrix Krylov subspaces $K_m(A, V_1)$ and $K_m(A^\top, W_1)$, respectively and the following relations hold true:

$$A\mathcal{V}_m = \mathcal{V}_m * T_m + \delta_{m+1}[0_{n\times q}, \ldots, 0_{n\times q}, V_{m+1}], \tag{7.1.5}$$

$$A^\top \mathcal{W}_m = \mathcal{W}_m * T_m^\top + \beta_{m+1}[0_{n\times q}, \ldots, 0_{n\times q}, W_{m+1}], \tag{7.1.6}$$

$$\mathcal{W}_m^\top \diamond A\mathcal{V}_m = T_m. \tag{7.1.7}$$

7.2 Approximation of the transfer function

Model order reduction aims to replace a large-scale system by a system of lower dimensions that has almost the same features. The Krylov subspace based methods appear to be often used as one of the model reduction techniques.

Consider the large linear dynamical system $\mathcal{S}$, we aim to construct a reduced system $\mathcal{S}_m$, by applying the following change of variable: $x = \mathcal{V}_m * \tilde{x}$, with $\tilde{x} \in \mathbb{R}^m$.

Set $V_1 = \dfrac{B}{\|B\|_F}$ and let W_1 be a matrix that satisfies $< V_1, W_1 >_F = 1$. Adopting the global Lanczos procedure, the produced matrices $\mathcal{V}_m$ and $\mathcal{W}_m$ verifying the condition $\mathcal{W}_m^\top \diamond \mathcal{V}_m = I_m$, result in finding the lower dimensional dynamical system given by:

$$\mathcal{S}_m : \begin{cases} \dot{\tilde{x}}(t) = T_m \tilde{x}(t) + \tilde{B}u(t) \\ \tilde{y}(t) = \tilde{C} * \tilde{x}(t) \end{cases} \tag{7.2.1}$$

where $T_m = \mathcal{W}_m^\top \diamond A\mathcal{V}_m$, $\tilde{B} = \mathcal{W}_m^\top \diamond B$ and $\tilde{C} = C\mathcal{V}_m$. The associated transfer function is then expressed in this form:

$$H_m(s) = C_m \left(sI - A_m\right)^{-1} B_m, \tag{7.2.2}$$

with $A_m = T_m \otimes I_q$, $B_m = \tilde{B} \otimes I_q$ and $C_m = \tilde{C}$.

We notice that the approximate transfer function $H_m(s)$ given by (7.2.2) is a Schur complement and we have the following theorem:

Theorem 7.2.1 *Let $H_m(s)$ be the transfer function that approximates $H(s)$. If $sI - T_m$ is a block strongly nonsingular matrix, then $H_m(s)$ is a Schur complement and we can express it as follows:*

$$H_m(s) = H_{m-1}(s) + G_m \left((sI - T_m) \otimes I_q / (sI - T_{m-1}) \otimes I_q\right)^{-1} G_m' \tag{7.2.3}$$

where the matrices G_m and G'_m are given by:

$$G_m = \left(\begin{pmatrix} CV_m & CV_1 & \dots & CV_{m-1} \\ 0_q & & & \\ \vdots & & (sI - T_{m-1}) \otimes I_q & \\ 0_q & & & \\ -\beta_m I_q & & & \end{pmatrix} / \left(sI - T_{m-1}\right) \otimes I_q \right), \tag{7.2.4}$$

and

$$G'_m = \left(\begin{pmatrix} 0_q & 0_q & \dots & -\delta_m I_q \\ \|B\|_F I_q & & & \\ 0_q & & (sI - T_{m-1}) \otimes I_q & \\ \vdots & & & \\ 0_q & & & \end{pmatrix} / \left(sI - T_{m-1}\right) \otimes I_q \right) \tag{7.2.5}$$

Proof 7 *According to* (7.2.2)*, we express the approximate transfer function $H_m(s)$ as:*

$$H_m(s) = C\mathcal{V}_m \left(sI - (T_m \otimes I_q)\right)^{-1} \left(\left(\mathcal{W}_m^\top \diamond B \right) \otimes I_q \right) \tag{7.2.6}$$

$$= C\mathcal{V}_m \left((sI - T_m) \otimes I_q\right)^{-1} \left(\|B\|_F \, e_1^{(m)} \otimes I_q \right) \tag{7.2.7}$$

$$= -\left(0_{p\times q} - C\mathcal{V}_m \left((sI - T_m) \otimes I_q\right)^{-1} \left(\|B\|_F \, e_1^{(m)} \otimes I_q \right) \right) \tag{7.2.8}$$

$$= -\left(\begin{pmatrix} 0_{p\times q} & CV_1 & \dots & CV_m \\ \|B\|_F I_q & & & \\ 0_q & & (sI - T_m) \otimes I_q & \\ \vdots & & & \\ 0_q & & & \end{pmatrix} / \left(sI - T_m\right) \otimes I_q \right) \tag{7.2.9}$$

$$= -\left(\begin{pmatrix} 0_{p\times q} & CV_1 & \dots & CV_{m-1} & CV_m \\ \|B\|_F I_q & & & & 0_q \\ 0_q & & (sI - T_{m-1}) \otimes I_q & & \vdots \\ \vdots & & & & -\beta_m I_q \\ 0_q & 0_q & \dots & -\delta_m I_q & (s - \alpha_m) I_q \end{pmatrix} / \left(sI - T_m\right) \otimes I_q \right). \tag{7.2.10}$$

Using the matrix Sylvester identity, $H_m(s)$ can be written as follows:

$$\begin{aligned} H_m(s) = {} & H_{m-1}(s) \\ & + \left(E_m / \left(sI - T_{m-1}\right) \otimes I_q\right) \left((sI - T_m) \otimes I_q / \left(sI - T_{m-1}\right) \otimes I_q \right)^{-1} \left(E'_m / \left(sI - T_{m-1}\right) \otimes I_q\right), \end{aligned} \tag{7.2.11}$$

where

$$E_m = \begin{pmatrix} CV_1 & \dots & CV_{m-1} & CV_m \\ & & & 0_q \\ & (sI - T_{m-1}) \otimes I_q & & \vdots \\ & & & 0_q \\ & & & -\beta_m I_q \end{pmatrix} \quad \textit{and} \quad E'_m = \begin{pmatrix} \|B\|_F I_q & & & \\ 0_q & & & \\ \vdots & & (sI - T_{m-1}) \otimes I_q & \\ 0_q & & & \\ 0_q & 0_q & \dots & -\delta_m I_q \end{pmatrix}. \tag{7.2.12}$$

It follows that:

$$(E_m / (sI - T_{m-1}) \otimes I_q) = \left(\begin{pmatrix} CV_m & CV_1 & \dots & CV_{m-1} \\ 0_q & & & \\ \vdots & & (sI - T_{m-1}) \otimes I_q & \\ 0_q & & & \\ -\beta_m I_q & & & \end{pmatrix} / (sI - T_{m-1}) \otimes I_q \right) = G_m, \tag{7.2.13}$$

and

$$(E'_m / (sI - T_{m-1}) \otimes I_q) = \left(\begin{pmatrix} 0_q & 0_q & \dots & -\delta_m I_q \\ \|B\|_F I_q & & & \\ 0_q & & (sI - T_{m-1}) \otimes I_q & \\ \vdots & & & \\ 0_q & & & \end{pmatrix} / (sI - T_{m-1}) \otimes I_q \right) = G'_m. \tag{7.2.14}$$

□

The matrices G_m and G'_m defined in Theorem 7.2.1 satisfy the following result.

Theorem 7.2.2 *Let G_m and G'_m be the matrices defined by* (7.2.4) *and* (7.2.5)*, respectively. If the matrix $sI - T_m$ is block strongly nonsingular, then:*

$$G_m = CV_m + \beta_m G_{m-1} \left((sI - T_{m-1}) \otimes I_q / (sI - T_{m-2}) \otimes I_q \right)^{-1}, \tag{7.2.15}$$

$$G'_m = \delta_m \left((sI - T_{m-1}) \otimes I_q / (sI - T_{m-2}) \otimes I_q \right)^{-1} G'_{m-1}. \tag{7.2.16}$$

Proof 8 *We notice that the matrices G_m and G'_m are Schur complements and they can be written*

as:

$$G_m = \left(\begin{pmatrix} CV_m & CV_1 & \dots & CV_{m-1} \\ 0_q & & & \\ \vdots & & (sI - T_{m-1}) \otimes I_q & \\ 0_q & & & \\ -\beta_m I_q & & & \end{pmatrix} / \left(sI - T_{m-1}\right) \otimes I_q \right), \tag{7.2.17}$$

$$= \left(\begin{pmatrix} CV_m & CV_1 & \dots & CV_{m-2} & CV_{m-1} \\ 0_q & & & & 0_q \\ \vdots & & (sI - T_{m-2}) \otimes I_q & & \vdots \\ 0_q & & & & -\beta_{m-1} I_q \\ -\beta_m I_q & 0_q & \dots & -\delta_{m-1} I_q & (s - \alpha_{m-1}) I_q \end{pmatrix} / \left(sI - T_{m-1}\right) \otimes I_q \right), \tag{7.2.18}$$

and

$$G'_m = \left(\begin{pmatrix} 0_q & 0_q & \dots & -\delta_m I_q \\ \|B\|_F I_q & & & \\ 0_q & & (sI - T_{m-1}) \otimes I_q & \\ \vdots & & & \\ 0_q & & & \end{pmatrix} / \left(sI - T_{m-1}\right) \otimes I_q \right), \tag{7.2.19}$$

$$= \left(\begin{pmatrix} 0_q & 0_q & \dots & 0_q & -\delta_m I_q \\ \|B\|_F I_q & & & & 0_q \\ \vdots & & (sI - T_{m-2}) \otimes I_q & & \vdots \\ 0_q & & & & -\beta_{m-1} I_q \\ 0_q & 0_q & \dots & -\delta_{m-1} I_q & (s - \alpha_{m-1}) I_q \end{pmatrix} / \left(sI - T_{m-1}\right) \otimes I_q \right). \tag{7.2.20}$$

According to the matrix Sylvester identity, we have:

$$G_m = CV_m +$$

$$\beta_m \left(\begin{pmatrix} CV_1 & \dots & CV_{m-2} & CV_{m-1} \\ & & & 0_q \\ (sI - T_{m-2}) \otimes I_q & & & \vdots \\ & & & 0_q \\ & & & -\beta_{m-1} I_q \end{pmatrix} / \left(sI - T_{m-2}\right) \otimes I_q \right) \left(\left(sI - T_{m-1}\right) \otimes I_q / \left(sI - T_{m-2}\right) \otimes I_q \right)^{-1} \tag{7.2.21}$$

$$= CV_m + \beta_m G_{m-1} \left(\left(sI - T_{m-1}\right) \otimes I_q / \left(sI - T_{m-2}\right) \otimes I_q \right)^{-1}, \tag{7.2.22}$$

$$G'_m =$$

$$\delta_m \left((sI - T_{m-1}) \otimes I_q / (sI - T_{m-2}) \otimes I_q\right)^{-1} \left(\begin{pmatrix} 0_q & 0_q & \dots & -\delta_{m-1} I_q \\ \|B\|_F I_q & & & \\ 0_q & & (sI - T_{m-2}) \otimes I_q & \\ \vdots & & & \\ 0_q & & & \end{pmatrix} / (sI - T_{m-2}) \otimes I_q \right) \tag{7.2.23}$$

$$= \delta_m \left((sI - T_{m-1}) \otimes I_q / (sI - T_{m-2}) \otimes I_q\right)^{-1} G'_{m-1}. \tag{7.2.24}$$

□

The theorem below is an immediate consequence of the matrix Sylvester identity.

Theorem 7.2.3 *At step m, we assume that $sI - T_m$ is a block strongly nonsingular matrix, then we can express the matrix $\left((sI - T_m) \otimes I_q / (sI - T_{m-1}) \otimes I_q\right)$ in the following form:*

$$\left((sI - T_m) \otimes I_q / (sI - T_{m-1}) \otimes I_q\right) = (s - \alpha_m) I_q - \delta_m \beta_m \left((sI - T_{m-1}) \otimes I_q / (sI - T_{m-2}) \otimes I_q\right)^{-1}. \tag{7.2.25}$$

Proof 9

$$\left((sI - T_m) \otimes I_q / (sI - T_{m-1}) \otimes I_q\right)$$

$$= \left(\begin{pmatrix} & & & 0_q \\ & (sI - T_{m-1}) \otimes I_q & & \vdots \\ & & & 0_q \\ & & & -\beta_m I_q \\ 0_q & \dots & 0_q \quad -\delta_m I_q & (s - \alpha_m) I_q \end{pmatrix} / (sI - T_{m-1}) \otimes I_q \right), \tag{7.2.26}$$

$$= \left(\begin{pmatrix} (s - \alpha_m) I_q & 0_q & \dots & 0_q & -\delta_m I_q \\ 0_q & & & & \\ \vdots & & (sI - T_{m-1}) \otimes I_q & & \\ 0_q & & & & \\ -\beta_m I_q & & & & \end{pmatrix} / (sI - T_{m-1}) \otimes I_q \right), \tag{7.2.27}$$

$$= \left(\begin{pmatrix} (s - \alpha_m) I_q & 0_q & \dots & 0_q & -\delta_m I_q \\ 0_q & & & & 0_q \\ \vdots & & (sI - T_{m-2}) \otimes I_q & & \vdots \\ 0_q & & & & -\beta_{m-1} I_q \\ -\beta_m I_q & 0_q & \dots & -\delta_{m-1} I_q & (s - \alpha_{m-1}) I_q \end{pmatrix} / (sI - T_{m-1}) \otimes I_q \right). \tag{7.2.28}$$

Applying the matrix Sylvester identity, we obtain:

$$\left((sI - T_m) \otimes I_q / (sI - T_{m-1}) \otimes I_q\right) = (s - \alpha_m) I_q - \delta_m \beta_m \left((sI - T_{m-1}) \otimes I_q / (sI - T_{m-2}) \otimes I_q\right)^{-1}. \tag{7.2.29}$$

□

Using the global Lanczos process and the three previous results, we introduce the new algorithm for approximating the transfer function:

Algorithm 7

1 Set $\gamma = \|B\|_F$, $V_1 = B/\gamma$ *and* $\tilde{C}_1 = CV_1$

2 Choose $W_1 \in \mathbb{R}^{n \times q}$ *such that* $< W_1, V_1 >_F = 1$

3 Set $G_1 = \tilde{C}_1$, $G_1' = \gamma I_q$ *and* $\beta_1 = \delta_1 = 0$

4 Set $W_0 = V_0 = 0$, $\left((sI - T_0) \otimes I_q / (sI - T_{-1}) \otimes I_q\right) = 0$ *and* $H(s) \equiv 0$

5 For $j = 1, 2, \ldots$ *Do*

- $\alpha_j = tr(W_j^\top A V_j)$
- $\tilde{V}_{j+1} = AV_j - \alpha_j V_j - \beta_j V_{j-1}$
- $\tilde{W}_{j+1} = A^\top W_j - \alpha_j W_j - \delta_j W_{j-1}$
- $\delta_{j+1} = |tr(\tilde{V}_{j+1}^\top \tilde{W}_{j+1})|^{1/2}$
- $\beta_{j+1} = tr(\tilde{V}_{j+1}^\top \tilde{W}_{j+1})/\delta_{j+1}$
- $V_{j+1} = \tilde{V}_{j+1}/\delta_{j+1}$
- $\tilde{C}_{j+1} = CV_{j+1}$
- $W_{j+1} = \tilde{W}_{j+1}/\beta_{j+1}$
- $\left((sI - T_j) \otimes I_q / (sI - T_{j-1}) \otimes I_q\right) = (s - \alpha_j) I_q - \delta_j \beta_j \left((sI - T_{j-1}) \otimes I_q / (sI - T_{j-2}) \otimes I_q\right)^{-1}$
- $H_j(s) = H_{j-1}(s) + G_j \left((sI - T_j) \otimes I_q / (sI - T_{j-1}) \otimes I_q\right)^{-1} G_j'$
- $G_{j+1} = \tilde{C}_{j+1} - \beta_{j+1} G_j \left((sI - T_j) \otimes I_q / (sI - T_{j-1}) \otimes I_q\right)^{-1}$
- $G_{j+1}' = \delta_{j+1} \left((sI - T_j) \otimes I_q / (sI - T_{j-1}) \otimes I_q\right)^{-1} G_j'$

End Do.

7.3 Numerical examples

In order to test the reliability of the algorithm 7, the first example considers a SISO system of order 200. Given the dynamical system (7.0.1) where the state-space matrices A, B and C are chosen

randomly of sizes $n \times n$, $n \times q$ and $p \times n$, respectively, with $p = q = 1$.

s	$H(s)$	$H_m(s)$	$\frac{\|H(s) - H_m(s)\|}{\|H(s)\|}$
0.1	$6.533910158717696\ .10^{7}$	$6.533910197690468\ .10^{7}$	$5.96\ .10^{-09}$
1	88.895032156779010	88.895032156779038	$3.19\ .10^{-16}$
10	4.833910416480292	4.833910416480292	$1.83\ .10^{-16}$
10^2	0.461081699369726	0.461081699369726	$2.40\ .10^{-16}$
10^3	0.045895770811674	0.045895770811674	$1.51\ .10^{-16}$
10^5	$4.587252077456183\ .10^{-4}$	$4.587252077456183\ .10^{-4}$	0

Table 7.1: Comparison of the exact transfer function H and the approximate transfer function H_m (SISO system).

In the second example, we consider a MIMO system of order 200, where the state-space matrices A, B and C are chosen randomly of sizes $n \times n$, $n \times q$ and $p \times n$, respectively, with $p = q = 2$.

s	$H(s)$	$H_m(s)$
0.1	$\begin{bmatrix} 1094196466.383238 & 0657035780.480597 \\ 1663204459.810768 & 1001176218.056392 \end{bmatrix}$	$\begin{bmatrix} 1094196787.392356 & 0657035927.126166 \\ 1663204949.140197 & 1001176441.258157 \end{bmatrix}$
1	$\begin{bmatrix} 102.8383938210619 & 103.1640812780378 \\ 118.2309104280803 & 118.1814831820834 \end{bmatrix}$	$\begin{bmatrix} 102.8383938210618 & 103.1640812780378 \\ 118.2309104280801 & 118.1814831820833 \end{bmatrix}$
10	$\begin{bmatrix} 5.205552861383067 & 5.187197376674791 \\ 5.757836148044022 & 5.535028447816985 \end{bmatrix}$	$\begin{bmatrix} 5.205552861383066 & 5.187197376674787 \\ 5.757836148044020 & 5.535028447816984 \end{bmatrix}$
10^2	$\begin{bmatrix} 0.497783910472145 & 0.495584448641896 \\ 0.549259818805492 & 0.525441779341035 \end{bmatrix}$	$\begin{bmatrix} 0.497783910472145 & 0.495584448641896 \\ 0.549259818805492 & 0.525441779341035 \end{bmatrix}$
10^3	$\begin{bmatrix} 0.049563392434554 & 0.049340468986459 \\ 0.054676907227190 & 0.052280088880533 \end{bmatrix}$	$\begin{bmatrix} 0.049563392434554 & 0.049340468986459 \\ 0.054676907227189 & 0.052280088880533 \end{bmatrix}$
10^5	$\begin{bmatrix} 0.000495398753840 & 0.000493166314807 \\ 0.000546496782016 & 0.000522512116250 \end{bmatrix}$	$\begin{bmatrix} 0.000495398753840 & 0.000493166314807 \\ 0.000546496782016 & 0.000522512116250 \end{bmatrix}$

Table 7.2: Comparison of the exact transfer function H and the approximate transfer function H_m (MIMO system).

s	0.1	1	10	10^2	10^3	10^5
$\frac{\|H(s)-H_m(s)\|}{\|H(s)\|}$	$2.67\ .10^{-07}$	$1.26\ .10^{-15}$	$4.71\ .10^{-16}$	$8.26\ .10^{-16}$	$5.83\ .10^{-16}$	$5.07\ .10^{-16}$

Table 7.3: Relative error between the exact transfer function H and the approximate transfer function H_m (MIMO system).

Table 7.1, Table 7.2 and Table 7.3 present the transfer function H and the approximate transfer function H_m for different values of s. The relative error is also given to show the good performance of the proposed algorithm for both SISO and MIMO systems.

Conclusion

In the present work, we attempt to approximate the transfer function using the Schur complements properties and the global nonsymmetric Lanczos procedure which belongs to the category of the Krylov based model reduction methods. Some numerical experiments are afforded to illustrate the efficacy and the accuracy of the presented approach.

Conclusion and future work

In the present work, we have been interested in the resolution of the Schrödinger equation by applying Sinc numerical methods combined with double exponential transformations. Chapter 2 was a double exponential version of a study that has been done previously where the authors applied the single exponential Sinc collocation and Sinc-Galerkin methods for the wave function of the two-dimensional time-dependent Schrödinger equation. In Chapter 3, we also utilized the DESCM and SESGM for the computation of the energy eigenvalues of the time-independent Schrödinger equation in two and three dimensions. We treated the special case where the potentials are separable and consequently the resolution of the resulting large system is transformed into that of two smaller eigenvalue problems. In the numerical experiments, we took advantage from the one dimensional study while selecting the appropriate conformal mappings as well as the coefficients in relation with the asymptotic behavior of the wave function. Besides, we compared the results obtained using DESCM and DESGM with the corresponding SESCM and SESGM and the numerical results illustrated clearly the high accuracy and efficacy of DESCM and DESGM. The Schrödinger equation with nonseparable potential functions has been numerically treated in Chapter 4. Given some particular parity assumptions on the potential function and the double exponential transformation, we realized that the discretized systems stemming form the DESinc methods are block centrosymmetric. In fact, we extended the well-known centrosymmetric property to a bidimensional space by working with block matrices as components of the total matrix. The computation time is reduced by half due to the block centrosymmetry.

On the other hand, our interest has been also focused on solving large sparse linear systems with the same coefficient matrix and different right-hand sides. Such systems result from the discretization of differential equations arising in several fields. In Chapter 5, we proposed a new variant of the block nonsymmetric Lanczos method (Bl-NLM) based on the block nonsymmetric Lanczos process. We established new expressions of the sequence of solutions in the form of Schur complements. The properties of the latter helped constructing recursive formulas that have been useful for the execution of the block nonsymmetric Lanczos method. The Schur complements have been also utilized in Chapter 6 for the implementation of a direct variant of the global nonsymmetric Lanczos method (Gl-NLM). We developed recurrence expressions basing on the matrix Sylvester identity in order to elaborate a new version of the global nonsymmetric Lanczos method based on the global nonsymmetric Lanczos process. We compared our numerical results with those obtained while using other block and global Krylov subspace methods. The block and global nonsymmetric Lanczos methods ensured a better performance for some matrices, in terms of the number of iterations as well as the memory storage required. Chapter 7 can be seen as an application of the new version of the global Lanczos method for the approximation of the transfer function. The global

nonsymmetric Lanczos procedure was used as a model order reduction technique to produce a smaller linear dynamical system with lower time and storage requirements. Whereas the use of the Schur complements helped expressing the transfer function in a recursive form which gives a new implementation of the approach.

For future work, we aim for the eventuality of gathering the two aspects described in the two parts constituting this thesis. More specifically, it would be worthwhile utilizing the double exponential Sinc methods for the discretization of other multi-dimensional differential equations and trying to solve the resulting large systems using Lanczos-based methods.

Bibliography

[1] J. Lund and K. L. Bowers. *Sinc Methods for Quadrature and Differential Equations*. SIAM, Philadelphia, PA, 1992.

[2] F. Stenger. *Numerical Methods based on Sinc and Analytic Functions*. Springer-Verlag, New York, 1993.

[3] D. V. Ouellette. Schur complements and statistics. *Linear Algebra and its Applications*, 36:187–295, 1981.

[4] R. A. Brualdi and H. Schneider. Determinantal identities: Gaussi, schur, cauchy, sylvester, kronecker, jacobi, binet, laplace, muir and cayley. *Linear Algebra and its Applications*, 52:769–791, 1983.

[5] D. Carlson. Matrix decompositions involving the schur complement. *SIAM Journal on Applied Mathematics*, 28:577–587, 1975.

[6] P. Gaudreau, R. M. Slevinsky, and H. Safouhi. The double exponential sinc collocation method for singular sturm-liouville problems. *Journal of Mathematical Physics*, 57:1–19, 2016.

[7] M. Sugihara and T. Matsuo. Recent developments of the sinc numerical methods. *Journal of Computational and Applied Mathematics*, 164-165:673–689, 2004.

[8] Y. Saad. *Iterative methods for solving sparse linear systems*. 2nd edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003.

[9] K. Jbilou, A. Messaoudi, and K. Tabaa. Some schur complement identities and applications to matrix extrapolation methods. *Linear Algebra and its Applications*, 392:195–210, 2004.

[10] J. Schur. Uber potenzreihen, die im innern des einheitskreises beschrankt sind. *Journal fur die reine und angewandte Mathematik*, 147:205–232, 1917.

[11] K. Jbilou and A. Messaoudi. Matrix recursive interpolation algorithm for block linear systems: Direct methods. *Linear Algebra and its Applications*, 294:137–154, 1999.

[12] A. El Guennouni, K. Jbilou, and H. Sadok. The block lanczos method for linear systems with multiple right-hand sides. *Applied Numerical Mathematics*, 51:243–256, 2004.

[13] F. Stenger. *Handbook of Sinc Numerical Methods*. CRC Press, London, 2010.

[14] A. Nurmuhammad, M. Muhammad, and M. Mori. Sinc-galerkin method based on the de transformation for the boundary value problem of fourth-order ode. *Journal of Computational and Applied Mathematics*, 206:17–26, 2007.

[15] K. Parand, M. Dehghan, and A. Pirkhedri. The use of sinc-collocation method for solving falkner-skan boundary-layer equation. *International Journal for Numerical Methods in Fluids*, 68:36–47, 2012.

[16] K. Abdella, X. Yu, and I. Kucuk. Application of the sinc method to a dynamic elasto-plastic problem. *Journal of Computational and Applied Mathematics*, 233:626–645, 2009.

[17] A. Saadatmandi and M. Dehghan. The use of sinc-collocation method for solving multi-point boundary value problems. *Communications in Nonlinear Science and Numerical Simulation*, 17:593–601, 2012.

[18] F. Stenger. A sinc-galerkin method of solution of boundary value problems. *Mathematics of Computation*, 33:85–109, 1979.

[19] K. M. McArthur, K. L. Bowers, and J. Lund. Numerical implementation of the sinc-galerkin method for second-order hyberbolic equations. *Numerical Methods for Partial Differential Equations*, 3:169–185, 1987.

[20] J. Lund. Symmetrization of the sinc-galerkin method for boundary value problems. *Mathematics of Computation*, 47:571–588, 1986.

[21] M. M. Tharwat, A. H. Bhrawy, and A. Yildirim. Numerical computation of eigenvalues of discontinuous sturm-liouville problems with parameter dependent boundary conditions using sinc method. *Numerical Algorithms*, 63:27–48, 2013.

[22] M. M. Tharwat. Sinc approximation of eigenvalues of sturm-liouville problems with a gaussian multiplier. *Calcolo*, 51:465–484, 2013.

[23] T. S. Carlson, J. Dockery, and J. Lund. A sinc-collocation method for initial value problems. *Mathematics of Computation*, 66:215–235, 1997.

[24] P. Amore. A variational sinc collocation method for strong-coupling problems. *Journal of Physics A: Mathematical and General*, 39:349–355, 2006.

[25] X. Wu, W. Kong, and C. Li. Sinc collocation method with boundary treatment for two-point boundary value problems. *Journal of Computational and Applied Mathematics*, 196:229–240, 2006.

[26] F. Stenger. Approximations via whittaker's cardinal function. *Journal of Approximation Theory*, 17:222–240, 1976.

[27] M. Jarratt, J. Lund, and K. L. Bowers. Galerkin schemes and the sinc-galerkin method for singular sturm-liouville problems. *Journal of Computational Physics*, 89:41–62, 1990.

[28] N. Eggert, M. Jarratt, and J. Lund. Sinc function computation of the eigenvalues of sturm-liouville problems. *Journal of Computational Physics*, 69:209–229, 1987.

[29] H. Takahasi and M. Mori. Double exponential formulas for numerical integration. *Publications of the Research Institute for Mathematical Sciences*, 9:721–741, 1974.

[30] M. Sugihara. Optimality of the double exponential formula-functional analysis approach. *Numerische Mathematik*, 75:379–395, 1997.

[31] M. Sugihara. Near optimality of the sinc approximation. *Mathematics of Computation*, 72:767–786, 2003.

[32] P. J. Gaudreau, R. M. Slevinsky, and H. Safouhi. Computing energy eigenvalues of anharmonic oscillators using the double exponential sinc collocation method. *Annals of Physics*, 360:520–538, 2015.

[33] T. Cassidy, P. Gaudreau, and H. Safouhi. On the computation of eigenvalues of the anharmonic coulombic potential. *Journal of Mathematical Chemistry*, 56:477–492, 2018.

[34] P. Gaudreau and H. Safouhi. Double exponential sinc-collocation method for solving the energy eigenvalues of harmonic oscillators perturbed by a rational function. *Journal of Mathematical Physics*, 58, 2017.

[35] M. Essaouini, B. Abouzaid, P. Gaudreau, and H. Safouhi. Computation of energy eigenvalues of the anharmonic coulombic potential with irregular singularities. *Numerical Algorithms*, 2020.

[36] A. Cantoni and P. Butler. Eigenvalues and eigenvectors of symmetric centrosymmetric matrices. *Linear Algebra and its Applications*, 13:275–288, 1976.

[37] A. L. Andrew. Eigenvectors of certain matrices. *Linear Algebra and its Applications*, 7:151–162, 1973.

[38] I. T. Abu-Jeib. Centrosymmetric matrices: properties and an alternative approach. *Canadian Applied Mathematics Quarterly*, 10:429–445, 2002.

[39] F. Z. Zhou. The solvability conditions for the inverse eigenvalue problems of centro-symmetric matrices. *Linear Algebra and its Applications*, 364:147–160, 2003.

[40] P. Gaudreau and H. Safouhi. Centrosymmetric matrices in the sinc collocation method for sturm-liouville problems. *Mathematical Modeling and Computational Physics*, 2016.

[41] D. Kosloff and R. Kosloff. A fourier method solution for the time dependent schrödinger equation as a tool in molecular dynamics. *Journal of Computational Physics*, 52:35–53, 1983.

[42] M. Subasi. On the finite-differences schemes for the numerical solution of two dimensional schrödinger equation. *Numerical Methods for Partial Differential Equations*, 18:752–758, 2002.

[43] Y. V. Kopylov, A. V. Popov, and A. V. Vinogradov. Application of the parabolic wave equation to x-ray diffraction optics. *Optics Communications*, 118:619–636, 1995.

[44] J. C. Wells, D. R. Schultz, P. Gavras, and M. S. Pindzola. Numerical solution of the time-dependent schrödinger equation for intermediate-energy collisions of antiprotons with hydrogen. *Physical Review A*, 54:593–604, 1996.

[45] M. Dehghan and A. Shokri. A numerical method for two-dimensional schrödinger equation using collocation and radial basis functions. *Computers and Mathematics with Applications*, 54:136–146, 2007.

[46] A. Mohebbi and M. Dehghan. The use of compact boundary value method for the solution of two-dimensional schrödinger equation. *Journal of Computational and Applied Mathematics*, 225:124–134, 2009.

[47] M. Dehghan and F. Emami-Naeini. The sinc-collocation and sinc-galerkin methods for solving the two-dimensional schrödinger equation with nonhomogeneous boundary conditions. *Applied Mathematical Modelling*, 37:9379–9397, 2013.

[48] J. C. Kalita, P. Chhabra, and S. Kumar. A semi-discrete higher order compact scheme for the unsteady two-dimensional schrödinger equation. *Journal of Computational and Applied Mathematics*, 197:141–149, 2006.

[49] X. Antoine, C. Besse, and V. Mouysset. Numerical schemes for the simulation of the two-dimensional schrödinger equation using non-reflecting boundary conditions. *Mathematics of Computation*, 73:1779–1799, 2004.

[50] R. Koc and D. Haydargil. Solution of the schrödinger equation with one and two dimensional double-well potentials. *Turkish Journal of Physics*, 28:161–167, 2004.

[51] L. Gr. Ixaru. New numerical method for the eigenvalue problem of the 2d schrödinger equation. *Computer Physics Communications*, 181:1738–1742, 2010.

[52] H. Taseli and R. Eid. Eigenvalues of the two-dimensional schrödinger equation with nonseparable potentials. *International Journal of Quantum Chemistry*, 59:183–201, 1996.

[53] S. H. Dong. The ansatz method for analyzing schrödinger's equation with three anharmonic potentials in d dimensions. *Foundations of Physics Letters*, 15:385–395, 2002.

[54] S. Ikhdair and R. Sever. Exact polynomial eigensolutions of the schrödinger equation for the pseudoharmonic potential. *Journal of Molecular Structure: THEOCHEM*, 806:155–158, 2007.

[55] M. N. Farizky, A. Suparmi, C. Cari, and M. Yunianto. Solution of three dimensional schrödinger equation for eckart and manning-rosen non-central potential using asymptotic iteration method. *Journal of Physics: Conference Series*, 776:012085, 2016.

[56] A. B. Cruse. Some combinatorial properties of centrosymmetric matrices. *Linear Algebra and its Applications*, 16:65–77, 1977.

[57] A. Cantoni and P. Butler. Properties of the eigenvectors of persymmetric matrices with applications to communication theory. *IEEE Transactions on communication*, 24:804–809, 1976.

[58] L. Datta and S. D. Morgera. On the reducibility of centrosymmetric matrices - applications in engineering problems. *Circuits Systems and Signal Process*, 8:71–96, 1989.

[59] I. T. Abu-Jeib. Centrosymmetric and skew-centrosymmetric matrices and regular magic squares. *New Zealand Journal of Mathematics*, 33:105–112, 2004.

[60] B. H. Bransden and C. J. Joachain. *Quantum Mechanics*. 2nd edition, Pearson Prentice Hall, Essex, UK, 2000.

[61] C. C. Chu, M. H. Lai, and W. S. Feng. The multiple point global lanczos method for mutiple-inputs multiple-outputs interconnect order reductions. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E89-A:2706–2716, 2006.

[62] D. L. Boley. Krylov space methods on state-space control models. *Circuits, Systems and Signal Processing*, 13:733–758, 1994.

[63] M. Esghir, Y. Louartassi, and N. Elalami. New algorithm for computing transfer function from state equation. *Applied Mathematical Sciences*, 7:1171–1182, 2013.

[64] M. Esghir. Birecursive interpolation algorithm: A formalism for solving systems of linear equations. *Applied Mathematical Sciences*, 7:1157–1169, 2013.

[65] H. A. van der Vorst. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13:631–644, 1992.

[66] D. P. O'Leary. The block conjugate gradient algorithm and related methods. *Linear Algebra and its Applications*, 29:293–322, 1980.

[67] A. El Guennouni, K. Jbilou, and H. Sadok. A block version of bicgstab for linear systems with multiple right-hand sides. *Electronic Transactions on Numerical Analysis*, 16:129–142, 2003.

[68] A. El Guennouni, K. Jbilou, and A. J. Riquet. Block krylov subspace methods for solving large sylvester equations. *Numerical Algorithms*, 29:75–96, 2002.

[69] A. H. Bentbib, M. El Guide, and K. Jbilou. Matrix krylov subspace methods for image restoration. *New Trends in Mathematical Sciences*, 3:136–148, 2015.

[70] K. Jbilou, H. Sadok, and A. Tinzefte. Oblique projection methods for linear systems with multiple right-hand sides. *Electronic Transactions on Numerical Analysis*, 20:119–138, 2005.

[71] K. Jbilou, A. Messaoudi, and H. Sadok. Global fom and gmres algorithms for matrix equations. *Applied Numerical Mathematics*, 31:49–63, 1999.

[72] M. Heyouni and K. Jbilou. Matrix krylov subspace methods for large scale model reduction problems. *Applied Mathematics and Computation*, 181:1215–1228, 2006.

[73] M. Esghir, O. Ibrihich, M. Essaouini, and S. El Hajji. Transfer function matrices of state-space models. *Applied Mathematical Sciences*, 9:935–948, 2015.

[74] H. Barkouki, A. H. Bentbib, M. Heyouni, and K. Jbilou. An extended nonsymmetric block lanczos method for model reduction in large scale dynamical systems. *Calcolo*, 55, 2018.

[75] A. H. Bentbib, M. El Guide, and K. Jbilou. The block lanczos algorithm for linear ill-posed problems. *Calcolo*, 54:711–732, 2017.

[76] Z. Bai, D. Day, and Q. Ye. Able: an adaptive block lanczos method for non-hermitian eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications*, 20:1060–1082, 1999.

[77] *https://math.nist.gov/MatrixMarket/.*

[78] A. H. Bentbib, M. El Guide, K. Jbilou, and L. Reichel. A global lanczos method for image restoration. *Journal of Computational and Applied Mathematics*, 300:233–244, 2016.

[79] H. Barkouki, A. H. Bentbib, and K. Jbilou. A matrix rational lanczos method for model reduction in large-scale first- and second-order dynamical systems. *Numerical Linear Algebra with Applications*, 24, 2016.

[80] A. C. Antoulas. *Approximation of Large-Scale Dynamical Systems.* Advances in Design and Control, SIAM, Philadelphia, 2005.

[81] P. Benner, S. Gugercin, and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57:483–531, 2015.

[82] S. Gugercin, A. C. Antoulas, and C. Beattie. H2 model reduction for large-scale linear dynamical systems. *SIAM Journal on Matrix Analysis and Applications*, 30:609–638, 2008.

[83] S. Gugercin and A. C. Antoulas. Model reduction of large-scale systems by least squares. *Linear Algebra and its Applications*, 415:290–321, 2006.

[84] R. W. Freund. Krylov-subspace methods for reduced-order modeling in circuit simulation. *Journal of Computational and Applied Mathematics*, 123:395–421, 2000.

Abstract

The current thesis work is presented in two parts. The first part is devoted to the solution of the Schrödinger equation using Sinc numerical methods. The nonstationary version of this equation is treated in two dimensions using the double exponential Sinc collocation and Sinc-Galerkin methods (DESCM and DESGM). The resolution of the stationary Schrödinger equation using the DESCM and DESGM methods is also discussed in two- and three- dimensional spaces. The matrices arising from the discretization of the equation using the DESinc methods are block centrosymmetric. This property is introduced as a two-dimensional extension of the well-known centrosymmetric property. The second part of this study is aimed at resolving large sparse linear systems with multiple right-hand sides using the block and global nonsymmetric Lanczos methods (Bl-NLM and Gl-NLM) which are introduced in new versions. A new expression of the solution is presented basing on some Schur complements identities and consequently some recursive formulas are occurred to give rise to new algorithms for implementing the block and global Lanczos methods. The combination of the global nonsymmetric Lanczos process and the Schur complements is also applied for the approximation of the transfer function.

Keywords: Double exponentiel Sinc methods, Schrödinger equation, Block centrosymmetry, Block and global Lanczos methods, Schur complement, Transfer function.

Résumé

Ce travail de thèse est présenté en deux parties. La première partie est consacrée à la résolution de l'équation de Schrödinger par les méthodes numériques Sinc. La version non stationnaire de cette équation est traitée en deux dimensions en utilisant les méthodes de collocation Sinc et Sinc-Galerkin exponentielles doubles (DESCM et DESGM). La résolution de l'équation de Schrödinger stationnaire à l'aide des méthodes DESCM et DESGM est également discutée en deux et trois dimensions. Les matrices issues de la discrétisation de l'équation à l'aide des méthodes DESinc sont centrosymétriques par blocs. Cette propriété est introduite comme une extension bidimensionnelle de la propriété connue de centrosymétrie. La deuxième partie de cette étude vise à résoudre de grands systèmes linéaires denses avec plusieurs seconds membres en utilisant les méthodes de Lanczos non symétrique par blocs (Bl-NLM) et globale (Gl-NLM) qui sont introduites dans de nouvelles versions. Une nouvelle expression de la solution est présentée en se basant sur certaines identités du complément de Schur et, par conséquent, certaines formules récursives sont apparues pour donner naissance à de nouveaux algorithmes pour l'implémentation des méthodes Bl-NLM et Gl-NLM. La combinaison du processus de Lanczos non symétrique global et des compléments de Schur est également appliquée pour l'approximation de la fonction de transfert.

Mots-clés : Méthode Sinc exponentielles doubles, Equation de Schrödinger, Centrosymmétrie par blocs, Méthode de Lanczos par blocs et globale, Complément de Schur, Fonction de transfert.