การศึกษาสมรรถนะในการแก้ปัญหาระบบสมการเชิงเส้นของโปรแกรม MATLAB สำหรับลูกโซ่มาร์คอฟขนาดใหญ่

The Study of MATLAB Capabilies in Solving Linear Equations for Large Scale Markov Chain

นุชรินทร์ ทิพยวรรณากร, วฐา มินเสน, กวินธร สัยเจริญ, ธารินี อินทรประเสริฐ
และพีรยุทธ์ ชาญเศรษฐิกุล
สาขาวิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

บทคัดย่อ

งานวิจัยนี้สนใจศึกษาเปรียบเทียบความสามารถในการหาคำตอบของระบบสมการเชิงเส้น $Ax = b$ สำหรับปัญหาลูกโซ่มาร์คอฟขนาดใหญ่บนโปรแกรม MATLAB ด้วยการจำลองเมตริกซ์เปลี่ยนสถานะอย่างสุ่ม ใน 3 ลักษณะ คือ 1) เมทริกซ์หนาแน่น(Dense Matrix) ซึ่งกำหนดให้มีขนาดเป็น 500, 1000, 1500, …, 5000 และมีจำนวนสมาชิกในเมทริกซ์ที่มีค่าศูนย์เป็น 99%, 95%, 90%, 75%, 50%, 25%, 10%, 5% และ 1% 2) เมทริกซ์มากเลขศูนย์(Sparse Matrix) และ 3) เมทริกซ์แถบที่มากเลขศูนย์(Band Matrix) ซึ่งทั้งสองลักษณะกำหนดให้มีขนาดเป็น 10, 100, …, 1000000 ทั้งนี้ในแต่ละกรณีที่ศึกษาทำการสุ่มซ้ำจำนวน 5 รอบ การหาคำตอบของปัญหาลูกโซ่มาร์คอฟนี้ฟังก์ชันที่ใช้ในการหาคำตอบแบ่งออกเป็น 2 กลุ่ม คือ 1. วิธีการคำนวณพีชคณิตโดยตรง(Direct method) ได้แก่ วิธีการจำกัดของเกาส์(Gauss Elimination : GE) , วิธีการแยกตัวประกอบ LU (LU Factorization : LU) และวิธีแยกตัวประกอบ QR (QR Factorization : QR) 2. วิธีการวนซ้ำ (Iterative Method) ได้แก่ ไบคอนจูเกตเกรเดียนต์(Biconjugate Gradient : BICG) , ไบคอนจูเกตเกรเดียนที่เสถียรแล้ว(Biconjugate Gradient Stabilized : BICGSTAB), คอนจูเกตเกรเดียนต์กำลังสอง (Conjugate Gradient Square : CGS) , วิธีแยกตัวประกอบ QR ที่ให้กำลังสองน้อยที่สุด (Least squares QR Factorization : LSQR) และ เศษตกค้างควอไซมินิมอล(Quasiminimal Residual : QMR) โดยเปรียบเทียบระยะเวลาที่ใช้ในการคำนวณหาคำตอบของแต่ละวิธี ซึ่งประมวลผลบนเครื่องคอมพิวเตอร์เพนเทียม 4 (Pentium 4) ความเร็ว 2.4 GHz หน่วยความจำสำรอง 1 GB พบว่าสำหรับเมทริกซ์หนาแน่นทุกขนาด และระดับความหนาแน่นทุกระดับ วิธี CGS สามารถคำนวณได้เร็วที่สุด สำหรับเมทริกซ์มากเลขศูนย์ พบว่าวิธี GE สามารถคำนวณได้เร็วที่สุด และสำหรับเมทริกซ์แถบมากเลขศูนย์ พบว่าวิธี LU สามารถคำนวณได้เร็วที่สุด

คำสำคัญ ลูกโซ่มาร์คอฟ , วิธีคำนวณเชิงเลข , โปรแกรม MATLAB

บทนำ

กระบวนการสโทแคสติก (Stochastic Process) เป็นตัวแบบความน่าจะเป็น (Probabilistic Model) ของระบบที่มีการเปลี่ยนแปลงเกิดขึ้นอย่างสุ่ม เช่น ระบบแถวคอยการให้บริการของธนาคาร จะพบว่าเวลาระหว่างการเข้ารับบริการของลูกค้าและเวลาให้บริการของพนักงานเป็นตัวแปรสุ่ม ทำให้การเปลี่ยนแปลงต่างๆ เกี่ยวกับแถวคอย เช่น เวลารอรับบริการของลูกค้าในแถวคอยจึงเป็นตัวแปรสุ่มด้วยเช่นกัน ถ้าระบบที่ต้องการศึกษามีขนาดใหญ่มาก (Large Scale System) เช่น ปัญหาการจัดการสินค้าคงคลังของห้างสรรพสินค้าขนาดใหญ่ซึ่งมีสินค้าหลากหลายขนาดและชนิดที่จะต้องจัดเก็บ สินค้าแต่ละชนิดมีระยะเวลาในการจัดเก็บหรือระยะเวลาในการจัดส่งที่แตกต่างกันหรือมีความไม่แน่นอนเกิดขึ้น รวมทั้งอาจเกิดปริมาณความต้องการสินค้าไม่แน่นอนขึ้นด้วย ดังนั้นการกำหนดจำนวนสินค้าคงคลังที่เหมาะสมจึงมีความซับซ้อนมากขึ้น และปัญหาลูกโซ่มาร์คอฟขนาดใหญ่ (Large Scale Markov Chain)

ปัญหาลูกโซ่มาร์คอฟเสนอโดย อันเดรย์ อันเดรเอียวิช มาร์คอฟ (Andrei Andreyevich Markov) โดยจัดรูปแบบปัญหาให้อยู่ในรูปของระบบสมการเชิงเส้นแล้วใช้หลักการพีชคณิตเพื่อหาค่าความน่าจะเป็นในระยะยาว (π) ของสถานะต่างๆ ของระบบ ปัจจุบันมีโปรแกรมคอมพิวเตอร์หลากหลายโปรแกรมมาช่วยในการหาคำตอบของระบบสมการเชิงเส้นด้วยวิธีพีชคณิต ซึ่งถ้าปัญหาขนาดเล็ก การหาคำตอบด้วยวิธีพีชคณิตจะทำได้ง่ายใช้เวลาคำนวณน้อย แต่ถ้าปัญหาขนาดใหญ่การหาคำตอบด้วยวิธีดังกล่าวจะใช้เวลาคำนวณมากขึ้น จึงมีการพยายามคิดค้นวิธีการเชิงเลข (Numerical Method) เพื่อหาคำตอบด้วยวิธีการวนซ้ำ (Iterative Method) หลากหลายวิธี เพื่อลดเวลาในการคำนวณหาคำตอบเป็นสำคัญ

งานวิจัยนี้มีวัตถุประสงค์หลักเพื่อศึกษาสมรรถนะของฟังก์ชันการหาคำตอบของระบบสมการเชิงเส้นด้วยวิธีพีชคณิตโดยตรง (Direct method) และวิธีวนซ้ำของคำสั่งพื้นฐานในโปรแกรม MATLAB เพื่อหาคำตอบของปัญหาลูกโซ่มาร์คอฟตั้งแต่ขนาดเล็ก จนถึงขนาดใหญ่ 3 ลักษณะ คือ 1) เมทริกซ์หนาแน่น (Dense Matrix) 2) ลักษณะเมทริกซ์มากเลขศูนย์ (Sparse Matrix) และ 3) ลักษณะเมทริกซ์แถบมากเลขศูนย์ (Band Matrix) โดยฟังก์ชันการหาคำตอบประกอบไปด้วยวิธีการทั้งหมด 8 วิธี เพื่อเปรียบเทียบเวลาในการคำนวณหาคำตอบว่ามีฟังก์ชันใดสามารถคำนวณหาคำตอบได้เร็วที่สุด โดยวิธีการคำนวณเหล่านี้จะกล่าวโดยละเอียดในหัวข้อถัดไป

ทฤษฎีที่เกี่ยวข้อง

การแก้ปัญหาระบบสมการเชิงเส้น $Ax = b$ ด้วยโปรแกรม MATLAB เพื่อหาค่าของตัวแปร x ซึ่งเป็นเวคเตอร์ขนาด $n \times 1$ เมื่อกำหนดให้ A เป็นค่าคงที่ในรูปเมทริกซ์จตุรัสมีขนาด $n \times n$ และ b เป็นเวคเตอร์ค่าคงที่ขนาด $n \times 1$ จำแนกวิธีแก้ปัญหาดังกล่าวออกเป็น 2 กลุ่มใหญ่ๆ คือ วิธีการคำนวณพีชคณิตโดยตรง และวิธีการวนซ้ำ โดยในแต่ละกลุ่มมีหลายวิธีการหาคำตอบแตกต่างกัน [7][10][11] ดังนี้

1. วิธีการคำนวณโดยตรง เป็นวิธีซึ่งคำนวณหาค่าของสมาชิกในเวคเตอร์ x แต่ละตัวได้โดยการจัดรูปเมทริกซ์ A เสียใหม่ ซึ่งมีรายละเอียดดังนี้

1.1 วิธีการกำจัดของเกาส์ (Gaussian Elimination : GE) จัดให้อยู่ในรูปเมทริกซ์แบบสามเหลี่ยมบน (Upper triangular matrix) โดยวิธีการดำเนินการตามแนวแถว (Row Operation) จาก $Ax = b \longrightarrow Ux = c$ เมื่อ U เป็นเมทริกซ์แบบสามเหลี่ยมบน ขนาด $n \times n$ และ c เป็นเวคเตอร์ค่าคงที่ขนาด $n \times 1$ จากนั้นแทนค่าแบบย้อนกลับ (Back Substitution) สำหรับสมาชิกในเวคเตอร์ x ทีละตัว

1.2 วิธีการแยกตัวประกอบ LU (LU Factorization : LU) จะแยกเมทริกซ์ A ให้อยู่ในรูปผลคูณของเมทริกซ์แบบสามเหลี่ยมบนกับเมทริกซ์แบบสามเหลี่ยมล่าง (Lower Triangular Matrix) โดยพิจารณาจาก $Ax = b \longrightarrow LUx = b \longrightarrow L(Ux) = b \longrightarrow Ly = b$ เมื่อ L และ U เป็นเมทริกซ์แบบสามเหลี่ยมล่างและบนตามลำดับ มีขนาด $n \times n$ และ y เป็นเวคเตอร์ค่าคงที่ขนาด $n \times 1$ ซึ่งเกิดจากการแทนค่าแบบย้อนกลับของ $Ux = y$ จากนั้นแทนค่าเพื่อหาค่าสมาชิกในเวคเตอร์ x ทีละตัว

1.3 วิธีการแยกตัวประกอบ QR (QR Factorization : QR) จะแยกเมทริกซ์ A ให้อยู่ในรูปผลคูณของเมทริกซ์เชิงตั้งฉาก (Orthogonal Matrix) กับเมทริกซ์แบบสามเหลี่ยมบน โดยพิจารณาจาก $Ax = b \longrightarrow QRx = b \longrightarrow Rx = Q^T b$ เมื่อ Q และ R เป็นเมทริกซ์เชิงตั้งฉากและเมทริกซ์แบบสามเหลี่ยมบนตามลำดับ มีขนาด $n \times n$ โดยอาศัยคุณสมบัติของเมทริกซ์เชิงตั้งฉาก คือ $Q^T = Q^{-1}$ $n \times 1$ จากนั้นแทนค่าแบบย้อนกลับสำหรับสมาชิกในเวคเตอร์ x ทีละตัว [6][8][9]

2. วิธีการวนซ้ำ เป็นวิธีซึ่งประมาณค่าของสมาชิกแต่ละตัวในเมทริกซ์โดยผ่านการคูณเมทริกซ์ค่าคงที่และเวกเตอร์ของ $\hat{x}$ ที่ประมาณค่าได้จากการทำซ้ำหลายๆ ครั้ง มีรายละเอียดดังนี้ [1][2]

2.1 ไบคอนจูเกตเกรเดียนต์ (Biconjugate Gradient : BICG) วิธี BICG เป็นวิธีการวนซ้ำที่พัฒนามาจากวิธีคอนจูเกตเกรเดียนต์ (Conjugate Gradient : CG) ซึ่งวิธี CG นั้น เป็นการแก้ปัญหาระบบสมการเชิงเส้น $Ax = b$ โดยวิธีการสุ่มค่า x ใดๆ บนพื้นที่คำตอบ จากนั้นจะใช้

วิธีการหาเวคเตอร์เชิงตั้งฉากของค่าเศษเหลือ r เพื่อให้ได้ทิศทางการลู่เข้าสู่คำตอบซึ่งแสดงด้วยเวคเตอร์ฉาย p โดยหลักการของวิธี CG [2] มีขั้นตอนดังต่อไปนี้

1) กำหนดค่า $r_0 = b - Ax_0, p_0 = r_0$

2) สำหรับ $j = 0, 1, \ldots$ จนกระทั่งเริ่มลู่เข้าสู่คำตอบเมื่อค่าเศษเหลือ r_{j+1} ต่ำสุด

3) $\alpha_j = (r_j, r_j)/(Ap_j, p_j)$ คือ สัมประสิทธิ์แสดงทิศทางการลู่เข้าสู่คำตอบโดยที่ r j เป็นเวคเตอร์เชิงตั้งฉาก ซึ่งมีเงื่อนไขจำเป็น คือ $(r_j - \alpha_j Ap_j, r_j) = 0$

4) คำนวณ $x_{j+1} = x_j + \alpha_j p_j$

5) หาค่าเศษเหลือ $r_{j+1} = r_j - \alpha_j Ap_j$

6) $\beta_j = (r_{j+1}, r_{j+1})/(r_j, r_j)$

7) $p_{j+1} = r_{j+1} + \beta_j p_j$

โดยที่ $(a,b) = a^T b$

จากวิธี CG ประกอบด้วยค่าของเวคเตอร์ที่ต้องทำการคำนวณ 4 ค่า คือ x, p, Ap และ r ส่วนวิธี BICG นั้น ใช้พื้นฐานการคำนวณเช่นเดียวกับวิธี CG แต่เพิ่มปัญหาคู่ควบ $A^T x^* = b^*$ ซึ่งมีการคำนวณค่า A^T เข้ามาพิจารณาร่วมด้วย ทำให้เกิดเวคเตอร์เชิงตั้งฉากเพิ่มขึ้นมาอีก 2 ชุด ที่มาจากเวคเตอร์ p_j^* และ r_j^* ของปัญหาคู่ควบ จึงเป็นที่มาของชื่อวิธี BICG ซึ่งเมื่อเปรียบเทียบประสิทธิภาพแล้ว วิธี BICG ให้ผลลัพธ์ใกล้เคียงกับวิธี CG ในกรณีที่ระบบมีลักษณะสมมาตร สำหรับระบบที่มีลักษณะไม่สมมาตร วิธี BICG มักถูกนำมาเปรียบเทียบกับวิธีการหาค่าเศษเหลือต่ำสุดแบบทั่วไป (Generalized Minimal Residual: GMRES) ซึ่งผลลัพธ์ที่ได้จากวิธี GMRES จะให้ผลลัพธ์ค่าเศษเหลือต่ำสุด แต่ต้องใช้พื้นที่หน่วยความจำมาก ในขณะที่วิธี BICG ใช้พื้นที่หน่วยความจำน้อยกว่า แต่ไม่สามารถคำนวณค่าเศษเหลือต่ำที่สุดได้

2.2 คอนจูเกตเกรเดียนต์กำลังสอง (Conjugate Gradient Square : CGS)

วิธีนี้พัฒนาขึ้น เพื่อต้องการหลีกเลี่ยงการคำนวณค่า A^T ที่ใช้ในวิธี BICG ซึ่งทำให้การลู่เข้าสู่คำตอบกระทำได้เร็วกว่าวิธี BICG วิธีการที่ทำให้ A^T หายไป โดยใช้วิธีพหุนามกำลังสอง (Squared Polynomials) ซึ่งจากเดิมในวิธี BICG คำนวณ $r^* = \phi_j(A^T) r_0^* \longrightarrow r' = \phi_j{}^2(A) r_0$ และเช่นเดียวกับค่า p^* โดยที่ ϕ_j คือ องศาพหุนาม j (Polynomial of Degree j) ซึ่งในวิธีนี้จะเพิ่มเวคเตอร์ช่วยเพิ่มอีก 2 ค่า คือ u_j และ q_j [2] ดังขั้นตอนต่อไปนี้

1) กำหนดค่า $r_0 = b - Ax_0$ และ r_0^*

2) กำหนดให้ $p_0 = u_0 = r_0$

3) สำหรับ j = 0, 1, 2 ... จนกระทั่งลู่เข้าสู่คำตอบ

4) $\alpha_j = (r_j, r_0^*)/(Ap_j, r_0^*)$

5) $q_j = u_j - \alpha_j Ap_j$

6) $x_{j+1} = x_j + \alpha_j (u_j + q_j)$

7) $r_{j+1} = r_j - \alpha_j A(u_j + q_j)$

8) $\beta_j = (r_{j+1}, r_0^*)/(r_j, r_0^*)$

9) $u_{j+1} = r_{j+1} + \beta_j q_j$

10) $p_{j+1} = u_{j+1} + \beta_j (q_j + \beta_j p_j)$

ซึ่งการคำนวณโดยวิธี CGS นี้ ถึงแม้จะมีความสามารถในการลู่เข้าสู่คำตอบได้เร็วกว่าวิธี BICG แต่คำตอบที่ได้ค่อนข้างขาดความแม่นยำ และมีโอกาสสูงที่การค้นหาคำตอบในครั้งแรกจะไม่ถูกต้อง โดยทั่วไปวิธี CGS จึงเหมาะสมกับปัญหาที่ไม่สามารถหาค่า A^T หรือคำนวณได้ยาก

2.3 ไบคอนจูเกตเกรเดียนที่เสถียรแล้ว (Biconjugate Gradient Stabilized : BICGSTAB)

เป็นวิธีการที่พัฒนาขึ้น เพื่อแก้ปัญหาระบบสมการเชิงเส้นแบบไม่สมมาตร และหลีกเลี่ยงปัญหาการลู่เข้าสู่คำตอบที่ไม่สม่ำเสมอจากการคำนวณด้วยวิธี CGS ซึ่งจากเดิมวิธี CGS คำนวณ $r_j' = \phi_j^2(A)r_0 \longrightarrow r_j' = \psi_j(A)\phi_j(A)r_0$ โดยที่ ψ_j คือ องศาพหุนามเช่นเดียวกับ ϕ_j แต่ในวิธี BICGSTAB ได้นิยาม ψ_j ขึ้นมาใหม่ เพื่อปรับพฤติกรรมการลู่เข้าสู่คำตอบให้มีความราบเรียบมากขึ้น โดยวิธี BICGSTAB [2] มีขั้นตอนดังต่อไปนี้

1) กำหนดค่า $r_0 = b - Ax_0$ และ r_0^*

2) กำหนดให้ $p_0 = r_0$

3) สำหรับ j = 0, 1, … จนกระทั่งลู่เข้าสู่คำตอบ

4) $\alpha_j = (r_j, r_j^*)/(Ap_j, r_0^*)$

5) $s_j = r_j - \alpha_j Ap_j$

6) $\omega_j = (As_j, s_j)/(As_j, As_j)$

7) $x_{j+1} = x_j + \alpha_j p_j + \omega_j s_j$

8) $r_{j+1} = s_j - \omega_j As_j$

9) $\beta_j = \dfrac{(r_{j+1}, r_0^*)}{(r_j, r_0^*)} \dfrac{\alpha_j}{\omega_j}$

10) $p_{j+1} = r_{j+1} + \beta_j (p_j - \omega_j Ap_j)$

ซึ่งวิธี BICGSTAB นี้มีความสามารถลู่เข้าสู่คำตอบได้เร็วพอกันกับวิธี CGS แต่จะมีพฤติกรรมการลู่เข้าที่สม่ำเสมอกว่า

2.4 วิธีเศษตกค้างควอไซมินิมอล (Quasi-Minimal Residual : QMR)

เป็นวิธีที่มีพฤติกรรมในการพัฒนาเป็นแนวทางเดียวกันกับวิธีการหาค่าเศษเหลือต่ำสุดแบบทั่วไป โดยอาศัยการลู่เข้าของเชิงมุมฉากคู่ (Biorthogonal) ในการหาคำตอบของระบบที่เมทริกซ์ไม่ลักษณะสมมาตร โดยอาศัยหลักการของ Lanczos ที่มีเวคเตอร์ค่าคงเหลือ $r_0 = b - AX_0$ โดยการประมาณการลู่เข้าของคำตอบคือ $X = X_0 + V_m y$ กำหนดให้ $b - Ax = b - A(x_0 + V_m y)$

โดยอาศัยความสัมพันธ์ของ Lanczos นั้นเพื่อจัดให้อยู่ในรูปนอร์มของเวคเตอร์เศษตกค้าง (norm of residual vector) จะได้ $\|b - Ax\| = \|V_{m+1}(\beta e_1 - \overline{T}_m y)\|_2$ ซึ่งถ้าคอลัมน์เวคเตอร์ของ V_{m+1} เป็นเวคเตอร์เชิงตั้งฉากปกติ (Orthonormal Vector) $\|b - Ax\| = \|(\beta e_1 - \overline{T}_m y)\|_2$ จากนั้นจัดรูปเพื่อหาค่าคงเหลือต่ำที่สุดในรูปฟังก์ชันของ y จะได้ $J(y) = \|(\beta e_1 - \overline{T}_m y)\|_2$ การคำนวณความสัมพันธ์โดยประมาณของ $x_0 + V_m y$ ได้ค่าคำตอบซึ่งเรียกว่า ค่าประมาณของ QMR [2] ซึ่งมีขั้นตอนในการหาดังนี้

1) คำนวณ $r_0 = b - Ax_0$ และ $\gamma_0 := \|r_0\|_2, w_1 := v_1 := r_0 / \gamma_1$

2) กำหนดให้ m = 1, 2, …. จนกระทั่งลู่เข้าสู่คำตอบ แล้วคำนวณ α_m, δ_{m+1} และ v_{m+1}, w_{m+1} ตามขั้นตอนต่อไปนี้

- กำหนดค่าเวคเตอร์ v_1, w_1 เช่น $(v_1, w_1) = 1$
- กำหนด Set $\beta_1 = \delta_1 \equiv 0, w_0 = v_0 \equiv 0$
- สำหรับ j = 1 to m ให้ทำขั้นตอนต่อไปนี้
  - คำนวณ $\alpha_j = (Av_j, w_j)$
  - คำนวณ $\hat{v}_{j+1} = Av_j - \alpha_j v_j - \beta_j v_{j-1}$
  - คำนวณ $\hat{w}_{j+1} = A^T w_j - \alpha_j w_j - \delta_j w_{j-1}$
  - คำนวณ $\delta_{j+1} = \left|\hat{v}_{j+1}, \hat{w}_{j+1}\right|^{1/2}$. ถ้า $\delta_{j+1} = 0$ หยุด
  - คำนวณ $\beta_{j+1} = (\hat{v}_{j+1}, \hat{w}_{j+1}) / \delta_{j+1}$
  - คำนวณ $w_{j+1} = \hat{w}_{j+1} / \beta_{j+1}$
  - คำนวณ $v_{j+1} = \hat{v}_{j+1} / \delta_{j+1}$
- ปรับปรุงคำตอบของวิธีการแยกส่วนประกอบ QR ของ $\overline{T}_m$ เช่น
  - ใช้ $\Omega_i, i = m-2, m-1$ ถึง คอลัมน์ที่ m ของ $\overline{T}_m$
  - คำนวณค่าสัมประสิทธิ์การหมุน c_m, s_m โดย

$$s_m = \frac{h_{m+1,m}}{\sqrt{(h_{mm}^{(m-1)})^2 + h_{m+1,m}^2}}, c_m = \frac{h_{mm}^{(m-1)}}{\sqrt{(h_{mm}^{(m-1)})^2 + h_{m+1,m}^2}}$$

- ใช้หมุนรอบแกน Ω_m ถึง $\overline{T}_m$ และ $\overline{g}_m$ คำนวณ

3) $\gamma_{m+1} := -s_m \gamma_m$

4) $\gamma_m := c_m \gamma_m, and,$

5) $\alpha_m := c_m \alpha_m + s_m \delta_{m+1} (= \sqrt{\delta_{m+1}^2 + \alpha_m^2})$

6) $p_m = (v_m - \sum_{i=m-2}^{m-1} t_{im} p_i) / t_{mm}$

7) $x_m = x_{m-1} + \gamma_m p_m$

8) ถ้า $|\gamma_{m+1}| \le \varepsilon$ แล้ว หยุด

2.5 วิธีการแยกตัวประกอบ *QR* เพื่อกำลังสองน้อยที่สุด (Least-Square *QR* Factorization : *LSQR*)

จากสมการ $Ax = b$ จะใช้วิธี กำลังสองน้อยที่สุด (Least Squares) ช่วยในการหาคำตอบ จะได้สมการดังนี้ $Ax = b + r$, $A \in R^{m \times n}$ $A \in R^{mxn}$; $m \geq n$ เขียนได้ว่า $\|r\|^2 \longrightarrow \min_x \|Ax - b\|^2$ จากนั้นใช้การเปลี่ยนรูปวิธีของ Lanczos มีขั้นตอนดังนี้

$$\min_x \|Ax - b\|^2 \longrightarrow A = U_{k+1} \begin{bmatrix} B_k \\ 0 \end{bmatrix} V_k^T \longrightarrow \min_{a_k} \|B_k a_k - B_1 r_1\|^2 \longrightarrow Q_k B_k = \begin{bmatrix} R_k \\ 0^T \end{bmatrix}$$

$$\longrightarrow \hat{a}_k = R_k^{-1} Q_k (B_1 r_1) \longrightarrow \hat{x}_k = V_k \hat{a}_k \xrightarrow{truncation \quad criterion} \hat{x}, \|A\hat{x} - y\|^2$$

เมื่อเมทริกซ์ A ถูกแยกตัวประกอบเป็น U และ V ซึ่งเป็นเมทริกซ์เชิงตั้งฉากปรกติขนาด $(m \times (k+1))$ และ $(u \times k)$ ตามลำดับ และเป็นเมทริกซ์เชิงตั้งฉากปรกติ B_k ขนาด $(k + (1 \times k))$ โดย k เป็นจำนวนการวนซ้ำ และ ถ้า $k = u$ เมทริกซ์ A ก็จะถูกแยกตัวประกอบจนเต็มสมบูรณ์ เมื่อกำหนดให้ $\hat{a}_k$ เป็นสัมประสิทธิ์ของผลรวมเชิงเส้น

Q_k เป็นเมทริกซ์เชิงตั้งฉากปรกติที่แยกตัวประกอบมาจากเมทริกซ์ B_k

R_k เป็นเมทริกซ์แบบสามเหลี่ยมบนที่แยกตัวประกอบจากเมทริกซ์ B_k [13]

สำหรับปัญหาลูกโซ่มาร์คอฟ เป็นการแก้ปัญหาระบบสมการเชิงเส้น ซึ่งกำหนดได้จากคุณสมบัติของมาร์คอฟ โดยมีเมทริกซ์เปลี่ยนสถานะ (Transition Matrix) เขียนแทนด้วย P ซึ่งเป็นเมทริกซ์จัตุรัสขนาด $n \times n$ เพื่อหาค่าความน่าจะเป็นในระยะยาวของตัวแปรสถานะ (Steady State Probability) ซึ่งเป็นเวคเตอร์ขนาด $n \times 1$ เขียนแทนด้วย π มีรูปแบบความสัมพันธ์ดังนี้

$$\begin{aligned} \pi^T &= \pi^T P \\ 1 &= \pi^T \mathbf{1} \end{aligned} \tag{1}$$

จากนั้นจัดรูปแบบปัญหาให้อยู่ในรูป $Ax = b$ ได้ดังนี้

$$\begin{aligned} \pi^T (I - P) &= \mathbf{0} \\ \pi^T \mathbf{1} &= 1 \end{aligned} \tag{2}$$

เมื่อ I เป็นเมทริกซ์เอกลักษณ์ (Identity Matrix) ขนาด $n \times n$

เนื่องจากระบบนี้มีตัวแปร n ตัว แต่มีสมการ $n+1$ จึงสามารถตัดสมการที่ซ้ำซ้อน (Redundant) ได้หนึ่งสมการ ดังนั้นเมทริกซ์ A เวคเตอร์ x และ b จะมีลักษณะดังนี้

$$A = \begin{bmatrix} (I - P)_{(n-1)\times(n-1)} \\ \mathbf{1}_{1 \times n} \end{bmatrix}, \quad x = \pi \quad \text{และ} \quad b = \begin{bmatrix} \mathbf{0}_{(n-1)\times 1} \\ 1 \end{bmatrix}$$

จะเห็นได้ว่าการจัดรูปแบบลักษณะนี้จะทำให้ A เมทริกซ์จัตุรัสที่มีลักษณะไม่สมมาตรและไม่เป็นบวกแน่นอน (Non-Symmetric, Non-Positive Definite) [3][4][12]

วิธีการวิจัย

การวิจัยนี้จัดทำขึ้นเพื่อศึกษาสมรรถนะของโปรแกรม MATLAB ในการหาคำตอบสำหรับระบบสมการเชิงเส้นในปัญหาลูกโซ่มาร์คอฟขนาดใหญ่ บนเครื่องคอมพิวเตอร์เพนเทียม 4 (Pentium 4) ความเร็ว 2.4 GHz หน่วยความจำสำรอง 1 GB โดยมีขั้นตอนในการวิจัยดังนี้

1. กำหนดเมทริกซ์เปลี่ยนสถานะ P โดยสุ่ม ขนาด $n \times n$ โดยแบ่งออกเป็น 3 ลักษณะ คือ

1) มีลักษณะเป็นเมทริกซ์หนาแน่น โดยกำหนดระดับความหนาแน่นในแต่ละสถานะ 9 ระดับ คือ 1%, 5%, 10%, 25%, 50%, 75%, 90%, 95% และ 99% ตามลำดับ ทั้งนี้ขนาดของเมทริกซ์เปลี่ยนสถานะ (หรือจำนวนตัวแปรสถานะ) มีค่าเป็น $n = 500, 1000, 1500, \ldots, 5{,}000$

2) มีลักษณะเป็นเมทริกซ์มากเลขศูนย์ ที่ตัวแปรสถานะทุกตัวจะเปลี่ยนแปลงไปสู่สถานะที่อยู่ติดกันทางขวามือ และไม่เปลี่ยนแปลงสถานะด้วยความน่าจะเป็นคงที่ในแต่ละปัญหา กล่าวคือในตัวแปรสถานะจะเปลี่ยนแปลงไปได้ 2 สถานะเท่านั้น โดยที่ขนาดของ เมทริกซ์เปลี่ยนสถานะ มีค่าเป็น $n =$ 10, 100, 1000, …, 1000000

3) มีลักษณะเป็นเมทริกซ์แถบที่มากเลขศูนย์ ที่ตัวแปรสถานะทุกตัวจะเปลี่ยนแปลงไปสู่สถานะที่อยู่ติดกันทางขวามือและซ้ายมือ และไม่เปลี่ยนแปลงสถานะด้วยความน่าจะเป็นคงที่ในแต่ละปัญหา กล่าวคือในตัวแปรสถานะจะเปลี่ยนแปลงไปได้ 3 สถานะเท่านั้น โดยที่ขนาดของเมทริกซ์เปลี่ยนสถานะ มีค่าเป็น $n =$ 10, 100, 1000, …, 1000000

2. สร้างเมทริกซ์ A และ b ให้สอดคล้องกับสมการ (2)

3. สร้างชุดคำสั่งบนโปรแกรม MATLAB เพื่อหาคำตอบของระบบสมการเชิงเส้นด้วยวิธีต่างๆ ดังที่ได้กล่าวมาแล้ว บันทึกเวลาที่ใช้ในการประมวลผลของแต่ละวิธี [5][10]

4. วิเคราะห์ผล โดยเปรียบเทียบค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานของเวลาที่ใช้ในการประมวลผลแต่ละวิธี

5. สรุปผลการทดลอง

ผลการวิจัย (Analysis Result)

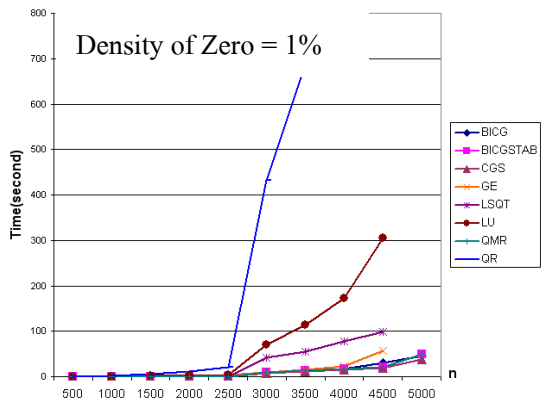
ผลการวิจัยแบ่งเป็น 3 ส่วนตามลักษณะของเมทริกซ์เปลี่ยนสถานะ ดังนี้

1. ลักษณะเมทริกซ์หนาแน่น ได้ผลการวิเคราะห์ดังรูปที่ 1 โดยพบว่า

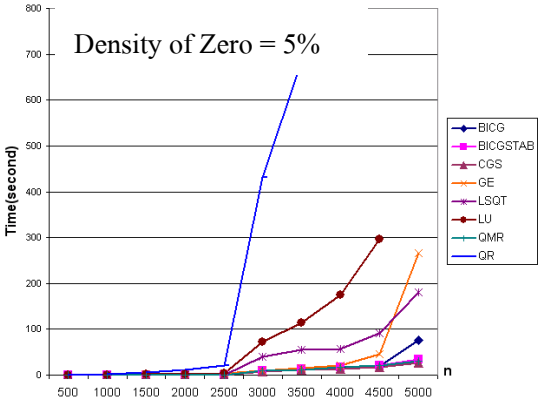
1) วิธี CGS สามารถหาคำตอบได้เร็วที่สุดเสมอ สำหรับเมทริกซ์เปลี่ยนสถานะทุกขนาด และระดับความหนาแน่นทุกระดับ

2) สำหรับเมทริกซ์เปลี่ยนสถานะขนาด 3,000-4,000 ซึ่งมีระดับความหนาแน่น 75% ขึ้นไป วิธีที่เร็วเทียบเท่ากับวิธี CGS คือ วิธี QMR, BICGSTAB และ BICG ตามลำดับ

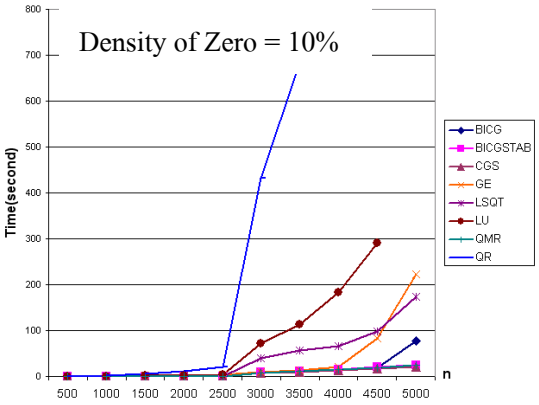
3) สำหรับเมทริกซ์เปลี่ยนสถานะขนาด 4,000 และ 5,000 ในทุกระดับความหนาแน่น วิธีที่เร็วเทียบเท่ากับวิธี CGS คือ วิธี QMR, BICGSTAB และ BICG ตามลำดับ



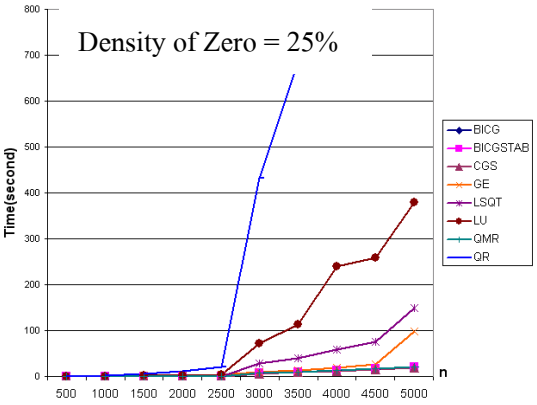
(1)



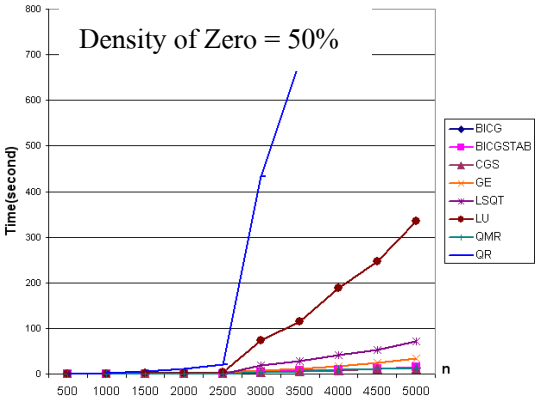
(2)



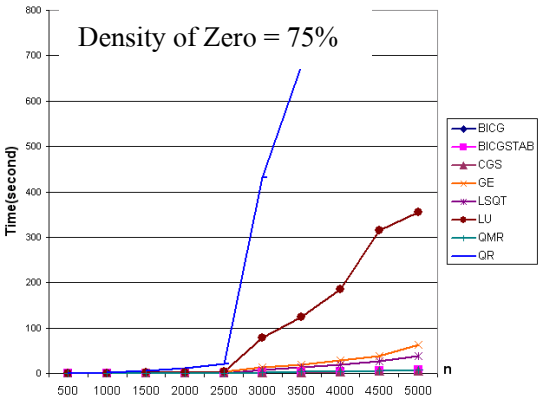
(3)



(4)

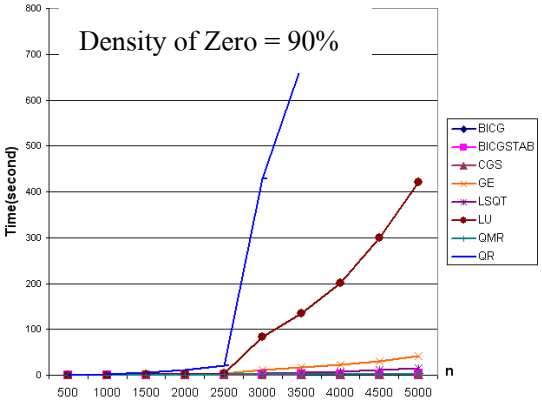


(5)

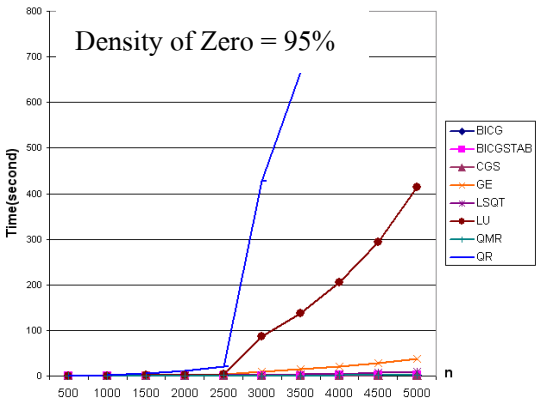


(6)

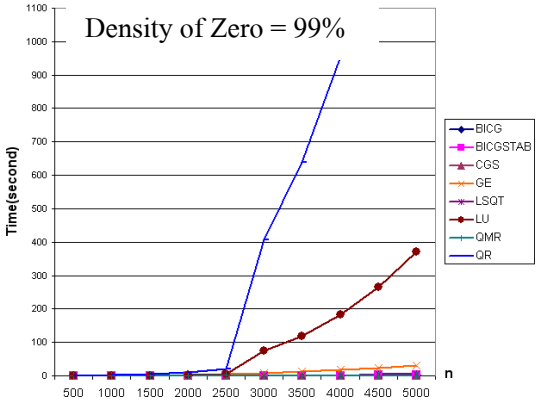
รูปที่1 แผนภาพเปรียบเทียบเวลาที่ใช้การคำนวณของแต่ละวิธี
โดยจำแนกตามระดับความหนาแน่นของเมทริกซ์



(7)



(8)



(9)

รูปที่1 (ต่อ) แผนภาพเปรียบเทียบเวลาที่ใช้การคำนวณของแต่ละวิธี
โดยจำแนกตามระดับความหนาแน่นของเมทริกซ์

2. ลักษณะเมทริกซ์มากเลขศูนย์ ได้ผลการวิเคราะห์ดังรูปที่ 2 โดยพบว่า

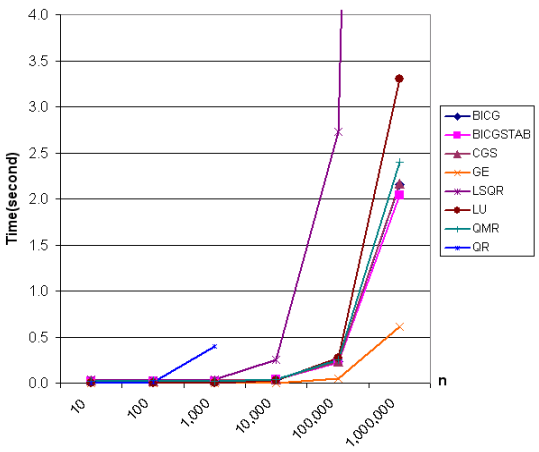
1) วิธี GE สามารถคำนวณได้เร็วที่สุดเสมอ สำหรับเมทริกซ์เปลี่ยนสถานะทุกขนาด

2) สำหรับเมทริกซ์เปลี่ยนสถานะ ที่มีขนาดไม่เกิน 10,000 วิธี LU สามารถคำนวณได้เร็วเทียบเท่าวิธี GE

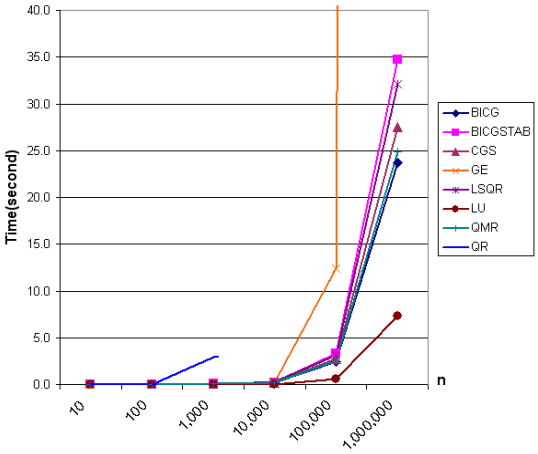
3. ลักษณะเมทริกซ์แถบมากเลขศูนย์ ได้ผลการวิเคราะห์ดังรูปที่ 3 โดยพบว่า

1) วิธี LU สามารถคำนวณได้เร็วที่สุดเสมอ สำหรับเมทริกซ์เปลี่ยนสถานะทุกขนาด

2) สำหรับเมทริกซ์เปลี่ยนสถานนะที่มีขนาดไม่เกิน 1,000 วิธี GE, CGS, BICG, BICGSTAB, LSQT, QR และ QMR จะเป็นวิธีที่สามารถคำนวณได้เร็วเทียบเท่ากับ LU ตามลำดับ



รูปที่2 แผนภาพเปรียบเทียบเวลาที่ใช้การคำนวณของแต่ละวิธีของลักษณะเมทริกซ์มากเลขศูนย์



รูปที่3 แผนภาพเปรียบเทียบเวลาที่ใช้การคำนวณของแต่ละวิธีของลักษณะเมทริกซ์แถบมากเลขศูนย์

สรุปผลการวิจัย

จากผลการวิเคราะห์พบว่า สำหรับเมทริกซ์หนาแน่นทุกขนาด และระดับความหนาแน่นทุกระดับ วิธี CGS สามารถคำนวณได้เร็วที่สุด สำหรับเมทริกซ์มากเลขศูนย์ พบว่าวิธี GE สามารถคำนวณได้เร็วที่สุด และสำหรับเมทริกซ์แถบมากเลขศูนย์ พบว่าวิธี LU สามารถคำนวณได้เร็วที่สุด

ในงานวิจัยนี้ มีข้อจำกัดอยู่ที่การพิจารณาเมทริกซ์ขนาดใหญ่สุดได้เพียง 5,000 x 5,000 ซึ่งในกรณีที่มีเมทริกซ์ขนาดใหญ่กว่านี้ หน่วยความจำสำรองที่มีอยู่ (RAM 1 GB) จะไม่สามารถทำการคำนวณได้ อาจใช้วิธีการแก้ปัญหาโดยการแบ่งส่วนเมทริกซ์ (Matrix Partitioning) เพื่อให้สามารถคำนวณเมทริกซ์ที่มีขนาดใหญ่ขึ้น โดยใช้หน่วยความจำสำรองขนาดเท่าเดิม

เอกสารอ้างอิง

[1] Leader, Jeffery L., **Numerical Analysis and Scientific Computation,** Pearson Education Inc. USA, 2004.

[2] Saad, Y., **Iterative methods for sparse linear system**, PWS Publishing, USA, 1996.

[3] Lasdon, Leon S. **Optimization Theory for Large Systems**, Macmillan Publishing. NewYork , 1970.

[4] Kulkarni, Vidyadhar G., **Modeling and Analysis of Stochastic Systems**, Chapmall & Hall, London ,1995.

[5] Stanoyevitch, Alexander**. Introduction to MATLAB with Numerical Preliminaries**. John Wiley & Sons, New Jersey, 2004.

[6] Steinberg, David I., **Computational Matrix Algebra**, McGraw-Hill Kogakusha, Japan, 1974.

[7] Leon, Steven J., **Linear Algebra with Applications**, 4th ed., Prentice Hall, New York, 1994.

[8] Schatzman, Michelle, **Numerical Analysis A Mathematical Introduction**. Oxford University Press, USA 2002.

[9] Strang, Gilbert, Borre, Kai. **Linear Algebra, Geodesy, and GPS**. Wellesley-Cambridge Press, Massachusetts, 1997.

[10] Chapman, Stephen J. **MATLAB Programming for Engineers**. 2nd ed., Brook & Cole. Canada, 2002
[11] Goldberg, Jack L. **Matrix Theory with Applications**. McGraw-Hill, Singapore 1991.
[12] Taylor, Howard, M., Karlin, Samuel, **An Introduction to Stochastic Modeling**. 3rd ed. Academic Press, USA 1998.
[13] Baur, Oliver., **Efficient Geopotential Recovery based on LSQR**. Geodetic Institute,Germany.