دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

الگوریتم QMR
(Quasi-Minimal Residual Algorithm)

سارینا حشمتی
دانشکده ریاضی و علوم کامپیوتر-رشته‌ی علوم کامپیوتر
پروژه‌ی کلاس جبرخطی‌عددی
خرداد ماه ۱۴۰۲

sarinaheshmatii@gmail.com

sarinaheshmati@aut.ac.ir

- **مقدمه**

در دنیای امروز، با پیشرفت فناوری، انتظار می‌رود که سیستم‌های هوشمند و موتورهای جستجوی پیشرفته، به سؤالات کاربران با سرعت و دقت بالا پاسخ دهند. در این راستا، الگوریتم QMR به عنوان یک الگوریتم ساده و کارآمد در حل مسائل پرسش و پاسخ، مورد استفاده قرار می‌گیرد. در این مقاله، به بررسی دقیق الگوریتم QMR و کاربرد آن در حل مسائل پرداخته خواهد شد. همچنین، در این مقاله، به مزایا و معایب استفاده از الگوریتم QMR ، پرداخته خواهد شد.

- QMR چیست؟

روش Quasi-Minimal Residual (QMR) یکی از روش‌های تکراری زیرفضای کرایلوف برای حل سیستم‌های معادلات خطی بزرگ و تنک است. این روش یک گسترش از روش Minimal (MINRES Residual) است که برای حل ماتریس‌های متقارن مثبت معین طراحی شده است.

روش QMR می‌تواند با ماتریس‌های غیرمتقارن و نامعین، همچنین ماتریس‌های منفرد و نزدیک به منفرد نیز مقابله کند. این روش به خصوص در صورتی که ماتریس دارای شرط شرایط بالا باشد یا سیستم دارای چندین بردار راست دست باشد، بسیار مؤثر است.

روش QMR دو زیرفضای کرایلوف را برای سیستم و سیستم همجوار آن ایجاد می‌کند. سپس این روش یک فرآیند دوطرفه‌سازی برای این زیرفضاها اعمال می‌کند تا یک جفت پایه دوطرفه‌سازی ایجاد شود. سپس با استفاده از یک رابطه بازگشتی دو مرحله‌ای، حل بدست می‌آید که با کمینه کردن نرم باقیمانده (residual norm) به صورت دوطرفه‌سازی انجام می‌شود.

یکی از مزایای روش QMR این است که نیازی به ذخیره ماتریس به صورت صریح ندارد، که باعث می‌شود برای مسائل با ماتریس‌های بزرگ و پراکنده، مناسب باشد. علاوه بر این، می‌توان این روش را با تکنیک‌های خوش حالت کنندگی ترکیب کرده تا همگرایی آن را شتاب دهیم.

با این حال، مانند سایر روش‌های فضای زیرمجموعه کریلوف، روش QMR ممکن است از نظر پایداری عددی دچار مشکل شود، به خصوص در صورتی که ماتریس بدوضع داشته باشد یا سیستم دارای چندین بردار راست دست باشد. به همین دلیل، پیاده‌سازی دقیق و نظارت دقیق بر روش ضروری است تا از قابلیت اعتماد و کارآیی آن اطمینان حاصل شود.

این روش به روش CG در این شباهت دارد که یک دنباله از بردارهایی را ایجاد می‌کند که یک فضای زیرمجموعه کریلوف تشکیل می‌دهند، اما برای شتاب دادن همگرایی و دقت بیشتر از دو جهت جستجو به جای یک جهت استفاده می‌کند.

الگوریتم QMR از طریق ایجاد دو زیرفضای کریلوف، یکی برای سیستم خوش‌حالت شده چپ و دیگری برای سیستم خوش‌حالت شده راست کار می‌کند. سپس سعی می‌کند یک بردار حل پیدا کند که همزمان به حداقل رساندن باقیمانده در هر دو زیرفضا کمک کند.

- **روش‌های زیرفضای کرایلف**

روش فضای زیرفضای کریلوف (Krylov Subspace Method) یک روش عددی برای حل معادلات خطی است که معمولاً برای ماتریس های بزرگ و تنک استفاده می شود. در این روش، فضای زیرفضای کریلوف که توسط یک ماتریس نامشخص تعریف می شود، به عنوان فضای جستجوی برای یافتن بردارهای پاسخ استفاده می شود.

برای حل یک معادله خطی با استفاده از روش فضای زیرفضای کریلوف، ابتدا یک بردار شروع را انتخاب می کنیم و سپس با استفاده از ماتریس مورد نظر و این بردار، یک فضای زیرفضای کریلوف را تشکیل می دهیم. سپس از روش‌های تقریبی برای یافتن بردارهای ارزش ویژه و یا حل معادلات خطی مورد نظر استفاده می شود.

مزیت این روش این است که فضای زیرفضای کریلوف با استفاده از تنها یک بردار شروع تشکیل می شود و همچنین روش‌های تقریبی برای یافتن ارزش ویژه استفاده می شوند، به صورت موازی قابل پیاده سازی هستند که این باعث می شود که این روش برای ماتریس های بزرگ و پراکنده مناسب باشد.

در مدل orthogonal ، برای تشکیل فضای زیرفضای کریلوف از روابط خطی بین بردارهای فضای زیرفضای کریلوف استفاده می‌شود. با استفاده از این روش، بردارهای فضای زیرفضای کریلوف به صورت متعامد هستند، یعنی در صورتی که دو بردار از فضای زیرفضای کریلوف را با هم ضرب داخلی کنیم، نتیجه برابر با صفر است.

در مدل bi-orthogonal ، برای تشکیل فضای زیرفضای کریلوف از روابط خطی بین بردارهای فضای زیرفضای کریلوف و بردارهای دیگری که از فضای مکمل به فضای زیرفضای کریلوف تشکیل شده‌اند، استفاده می‌شود. با استفاده از این روش، بردارهای فضای زیرفضای کریلوف و بردارهای مکمل به فضای زیرفضای کریلوف به صورت متعامد هستند، یعنی در صورتی که دو بردار از فضای زیرفضای کریلوف یا مکمل به آن را با هم ضرب داخلی کنیم، نتیجه برابر با صفر است.

در روش QMR ، از مدل bi-orthogonal برای تشکیل فضای زیرفضای کریلوف استفاده می‌شود. با استفاده از این روش، برای حل معادلات خطی، از یک ترکیب خطی از بردارهای فضای زیرفضای کریلوف و مکمل به آن استفاده می‌شود. این روش در مقایسه با روش‌های دیگر دقیق‌تر و پایدارتر است و برای ماتریس‌های بزرگ و پراکنده مناسب است.

- روش CG (conjugate gradient)

روش (Conjugate Gradient - CG) یک روش تکراری پرکاربرد برای حل سیستم‌های خطی بزرگ و تنک است. در مواردی که ماتریس متقارن و مثبت معین باشد.

روش CG با تولید یک دنباله از بردارهایی که یک فضای زیرفضا کریلوف را تشکیل می‌دهند، کار خود را آغاز می‌کند. این فضا به عنوان ترکیب خطی از ماتریس A و بردار شروع r_0 تعریف می‌شود، که در آن r_0 باقیمانده اولیه است. سپس این روش سعی می‌کند تا جواب x را پیدا کند که باقیمانده را در این فضا کمینه کند.

روش CG یک الگوریتم تکراری است که در هر تکرار، با جستجو برای بردار بعدی در فضای زیرمجموعه کریلوف که با تمام بردارهای قبلی هم‌راستایی دارد، جواب را بهبود می‌بخشد. این به این معناست که این روش تضمین می‌کند که جهت‌های جستجو با توجه به ماتریس A ارتگونال هستند، که منجر به همگرایی سریعتر نسبت به روش‌های تکراری دیگری می‌شود که از جهت‌های جستجوی غیرارتگونال استفاده می‌کنند.

روش CG می‌تواند به نمایش بیاورد که حداکثر در n بار تکرار، جواب دقیق را برای ماتریس A باشد، اگر A متقارن و مثبت معین باشد. در عمل، این روش معمولاً با سرعت بسیار بالاتری به هدف خود می‌رسد، به خصوص زمانی که با تکنیک‌های خوش‌حالت کنندگی که می‌توانند شرایط ماتریس را بهبود بخشند، ترکیب شوند.

یکی از مزایای روش CG این است که نیازی به ذخیره ماتریس به صورت صریح ندارد، که باعث می‌شود برای مسائل با ماتریس‌های بزرگ و پراکنده مناسب باشد. علاوه بر این، می‌توان آن را به راحتی به صورت موازی سازی کرد تا از معماری‌های محاسباتی مدرن استفاده کنیم.

با این حال، روش CG ممکن است از نظر پایداری عددی دچار مشکل شود، به خصوص زمانی که ماتریس بدوضع باشد یا سیستم دارای چندین بردار راست دست باشد. بنابراین، پیاده‌سازی دقیق و نظارت دقیق بر روش ضروری است تا از قابلیت اعتماد و کارآیی آن اطمینان حاصل شود.

- **الگوریتم Quasi-Minimal Residual (QMR)**

الگوریتم QMR می‌تواند به صورت زیر نوشته شود:

۱. یک بردار شروع x_0 را انتخاب کنید و $r_0 = b - A x_0$ را به عنوان باقیمانده اولیه تنظیم کنید.

۲. باقیمانده‌ی خوش‌حالت شده $M^{-1}r_0 = r_0$ را محاسبه کرده و جهت جستجوی اولیه $y_0 = N^{-1}z_0$ را تنظیم کنید.

۳. برای k از ۰ تا همگرایی:

الف) حاصل ضرب ماتریس-بردار Ay_k و $A^t z_k$ را محاسبه کنید.

ب) مقادیر α_k و β_k را با استفاده از فرمول‌های زیر محاسبه کنید:

$$\alpha_k = (r_k, z_k) / (y_k, Ay_k)$$

$$\beta_k = (r_k, A^t z_k) / (y_k, A^t z_k)$$

ج) جواب را با استفاده از فرمول زیر به روز کنید:

$$x_{k+1} = x_k + \alpha_k y_k$$

د) باقیمانده را با استفاده از فرمول زیر به روز کنید:

$$r_k = r_k - \alpha_k A y_k$$

ه) باقیمانده‌ی خوش‌حالت شده سمت چپ $M^{-1}r_{k+1} = z_{k+1}$ را محاسبه کنید.

و) باقیمانده‌ی پیش‌شرط‌گذاری شده سمت راست $N^{-1}A^t z_{k+1} = w_{k+1}$ را محاسبه کنید.

ز) با استفاده از فرمول‌های زیر جهت‌های جستجوی y_{k+1} و z_{k+1} را محاسبه کنید:

$$y_{k+1} = z_{k+1} + \beta_k y_k$$

$$z_{k+1} = w_{k+1} + \beta_k z_k$$

در اینجا، M و N به ترتیب خوش‌حالت کننده‌ی سمت چپ و راست هستند که برای ساده‌تر شدن سیستم‌های بدوضع مورد استفاده قرار می‌گیرند.

یکی از مزایای الگوریتم QMR این است که می‌تواند با ماتریس‌های نامعین یا منفرد که برای روش CG سخت هستند، کار کند. علاوه بر این، الگوریتم QMR می‌تواند برای ماتریس‌های بدوضع یا بسیار نامتقارن، مقاومت بیشتری نسبت به روش CG داشته باشد.

با این حال، الگوریتم QMR ممکن است از نظر هزینه محاسباتی گران‌تر از روش CG باشد، زیرا نیاز به محاسبه حاصل ضرب ماتریس-بردار Ay_k و $A^t z_k$ در هر بار تکرار را دارد که برای ماتریس‌های بزرگ ممکن است گران باشد. به علاوه، الگوریتم QMR ممکن است از نظر پایداری عددی، به ویژه وقتی که خوش‌حالت‌کننده‌ها به دقت انتخاب نشوند، مشکل داشته باشد. بنابراین، پیاده‌سازی دقیق و نظارت دقیق بر روش، برای اطمینان از قابلیت اطمینان و کارایی آن، ضروری است.

```python
import numpy as np

# define the matrix A
A = np.array([[2, -1, 0], [-1, 2, -1], [0, -1, 2]])

# define the right-hand side vector b
b = np.array([1, 0, 1])

# solve the system Ax = b using QMR
max_iter = 1000
tol = 1e-8
x = np.zeros_like(b, dtype=float)
r = b - A @ x
p = r.copy()
q = r.copy()
for k in range(max_iter):
    y = A @ q
    z = A.T @ p
    alpha = np.dot(r, q) / np.dot(y, z)
    beta = np.dot(r, z) / np.dot(y, z)
    x += alpha * p
    r -= alpha * y
    p = r + beta * p
    q = q - beta * z
    if np.linalg.norm(r) < tol:
        print(f"Converged in {k+1} iterations")
        break

# Print the solution
print("Converged in 1000 iterations \n")
print(f"Solution using QMR: {x}")
```

- **خوش‌حالت کننده‌ها (M و N)**

انتخاب خوش‌حالت کننده M و N می‌تواند تأثیر قابل توجهی بر همگرایی و کارآیی الگوریتم QMR داشته باشد. به طور کلی، هدف پیدا کردن خوش‌حالت کننده‌هایی است که سیستم‌های خوش‌حالت کننده‌ی مربوطه را ساده‌تر از سیستم اصلی قابل حل کنند.

چندین نوع خوش‌حالت کننده می‌توان با الگوریتم QMR استفاده کرد، از جمله ناقص تجزیه ومقیاس‌دهی قطری در ادامه چند نوع خوش‌حالت کننده رایج ذکر شده است:

۱ .ناقص تجزیه: تجزیه ناقص چولسکی (IC) و تجزیه ناقص LU (ILU) به ترتیب برای ماتریس‌های تقارنی و غیر تقارنی معمولاً به عنوان خوش‌حالت کننده استفاده می‌شوند.

۲ .مقیاس‌دهی قطری: این خوش‌حالت کننده به سادگی ماتریس A را با عناصر قطری آن مقیاس‌می‌کند که یک‌بار محاسبه و ذخیره می‌شوند. این مقدم شرط آسان برای پیاده‌سازی است و برای ماتریس‌های غالب قطری می‌تواند موثر باشد.

انتخاب خوش‌حالت کننده به خصوص وابسته به خصوصیات ماتریس A و مسئله خاصی است که در حال حل است. به طور کلی، یک خوش‌حالت کننده خوب باید دارای ویژگی‌های زیر باشد:

۱ .باید آسان قابل محاسبه و اعمال باشد.
۲ .باید عدد حالت را کاهش دهد.
۳ .باید ساختار و اسپارسیته ماتریس را به حداکثر حفظ کند.
۴ .باید مقاوم در مقابل تغییرات در ماتریس یا پارامترهای مساله باشد.

برای انتخاب خوش‌حالت کننده‌ها، می‌توان با انواع و پارامترهای مختلف مقدم شرط‌ها آزمایش کرده و عملکرد آنها را به روی یک مجموعه نماینده از مسائل تست کرد. همچنین می‌توان از مقدم شرط‌های تطبیقی یا ترکیبی استفاده کرد که نوع‌های مختلف مقدم شرط‌ها را ترکیب می‌کنند یا خوش‌حالت کننده را در طول تکرار بر اساس رفتار همگرایی تنظیم می‌کنند.

- **مزایا و معایب الگوریتم QMR**

الگوریتم QMR یک الگوریتم محاسباتی برای حل سیستم‌های معادلات خطی است که در برخی موارد می‌تواند از روش‌های دیگر مانند روش CG بهتر عمل کند. در ادامه مزایا و معایب الگوریتم QMR را بررسی می‌کنیم:

- مزایا:

۱. پایداری: الگوریتم QMR برای حل ماتریس‌های غیر هم‌تقارن یا نامعین، از روش CG پایدارتر است، زیرا در هر مرحله دو جهت جستجویی تولید می‌کند که می‌تواند با ویژگی‌های نامعین یا خصوصیت‌های ماتریس منفرد مدیریت کند.

۲. انعطاف پذیری: الگوریتم QMR با استفاده از خوش‌حالت‌کننده‌های مختلف، قابلیت تنظیم و تغییر دارد تا بهترین پیشرفت را در حل مسئله داشته باشد.

۳. کارایی: در مقایسه با روش‌های دیگر، الگوریتم QMR برای ماتریس‌های نامعین و غیر هم‌تقارن، ممکن است بهترین کارایی را داشته باشد.

- معایب:

۱. هزینه محاسباتی: الگوریتم QMR نیاز به محاسبه ضرب ماتریس-بردار در هر مرحله دارد که برای ماتریس‌های بزرگ، هزینه محاسباتی بالایی داشته باشد.

۲. نقص عددی: الگوریتم QMR ممکن است به دلیل استفاده از خوش‌حالت‌کننده‌های نامناسب و یا ماتریس دارای ارزش‌های ویژه بسیار بزرگ، با مشکلات ناشی از نقص عددی مواجه شود.

۳. حافظه: در مقابل روش‌های دیگر، الگوریتم QMR نیاز به حافظه بیشتری دارد، زیرا در هر مرحله دو جهت جستجویی و دو باقیمانده پیش شرط را به خاطر می‌سپارد.

- **کاربردهای الگوریتم QMR**

الگوریتم QMR در حل مسائل مختلف علوم ریاضیات، علم کامپیوتر و فیزیک مورد استفاده قرار می‌گیرد. برخی از کاربردهای این الگوریتم عبارتند از:

۱. حل معادلات دیفرانسیل با مشتقات جزئی: الگوریتم QMR در حل معادلات دیفرانسیل با مشتقات جزئی که معمولاً با ماتریس‌های نامعین مرتبط هستند، مورد استفاده قرار می‌گیرد.

۲. شبیه‌سازی فیزیکی: الگوریتم QMR در شبیه‌سازی فیزیکی مانند مکانیک سیالات، انتقال حرارت و الکترودینامیک مورد استفاده قرار می‌گیرد.

۳. تحلیل سیستم‌های قدرت: الگوریتم QMR در تحلیل سیستم‌های قدرت مانند شبکه‌های برق و سیستم‌های قدرت چند ناحیه‌ای مورد استفاده قرار می‌گیرد.

۴. فرآیندهای تصویربرداری: الگوریتم QMR در فرآیندهای تصویربرداری مانند بازسازی تصویر، تفکیک کامپوننت‌های مستقل و پردازش تصویر مورد استفاده قرار می‌گیرد.

۵. حل مسائل بهینه‌سازی: الگوریتم QMR در حل مسائل بهینه‌سازی مانند بهینه‌سازی برای کاربردهای مالی و بهینه‌سازی برای مسائل با ماتریس‌های نامعین مورد استفاده قرار می‌گیرد.

- **نتیجه‌گیری**

الگوریتم QMR یکی از الگوریتم‌های حل معادلات خطی با ماتریس نامعین و غیر متقارن است. این الگوریتم در سال ۱۹۹۱ توسط Freund و Nachtigal ارائه شد و در طی سال‌های بعدی به عنوان یکی از الگوریتم‌های پایدار و سریع در حل معادلات خطی مورد استفاده قرار گرفت.

از مزایای الگوریتم QMR می‌توان به پایداری آن در مقابل ماتریس‌های نامعین و غیر متقارن اشاره کرد. در این الگوریتم، در هر مرحله، دو جهت جستجویی تولید می‌شود که می‌تواند با ویژگی‌های نامعین و یا خصوصیت‌های ماتریس منفرد مدیریت کند. همچنین، این الگوریتم انعطاف پذیری بالایی دارد و با استفاده از پیش شرط‌های مختلف، قابلیت تنظیم و تغییر دارد تا بهترین پیشرفت را در حل مسئله داشته باشد.

از معایب الگوریتم QMR می‌توان به هزینه محاسباتی بالا و نقص عددی اشاره کرد. این الگوریتم نیاز به محاسبه ضرب ماتریس-بردار در هر مرحله دارد که برای ماتریس‌های بزرگ، هزینه محاسباتی بالایی دارد. همچنین، الگوریتم QMR ممکن است به دلیل استفاده از خوش‌حالت‌کننده‌های نامناسب و یا ماتریس دارای ارزش‌های ویژه بسیار بزرگ، با مشکلات ناشی از نقص عددی مواجه شود.

با توجه به بررسی مزایا و معایب الگوریتم QMR، می‌توان نتیجه گرفت که این الگوریتم در حل مسائلی که با ماتریس‌های نامعین و غیر هم‌تقارن مرتبط هستند، می‌تواند بهترین کارایی را داشته باشد. با این حال، هزینه محاسباتی بالا و نقص عددی می‌تواند مشکلاتی برای الگوریتم QMR ایجاد کند. به همین دلیل، برای استفاده از این الگوریتم، باید با دقت و با توجه به ویژگی‌های مسئله، پیش شرط‌های مناسب و معیارهای صحیحی را اعمال کرد تا از بهترین عملکرد الگوریتم QMR در حل مسئله بهره‌مند شد. در کل، الگوریتم QMR یکی از روش‌های مهم حل مسائل ماتریسی است که در بسیاری از زمینه‌های علمی و صنعتی مورد استفاده قرار می‌گیرد.

- **منابع**

۱. Yousef Saad "Iterative Methods for Sparse Linear Systems"

۲. https://mathworld.wolfram.com/Quasi-MinimalResidualMethod.html

۳. C. T. Kelley "Iterative methods for linear and nonlinear equations"