www.novin-eng.ir

آموزش کامل متد حل تکراری گوس-سایدل به کمک نرم افزار متلب (Gauss Seidel)

Gauss-Seidel Method

Differ from Jacobi method by sequential updating:
use new x_i immediately as they become available

$$\begin{cases} x_1^{new} = (b_1 - a_{12}x_2^{old} - a_{13}x_3^{old} - a_{14}x_4^{old}) / a_{11} \\ x_2^{new} = (b_2 - a_{21}x_1^{new} - a_{23}x_3^{old} - a_{24}x_4^{old}) / a_{22} \\ x_3^{new} = (b_3 - a_{31}x_1^{new} - a_{32}x_2^{new} - a_{34}x_4^{old}) / a_{33} \\ x_4^{new} = (b_4 - a_{41}x_1^{new} - a_{42}x_2^{new} - a_{43}x_3^{new}) / a_{44} \end{cases}$$

www.novin-eng.ir

متدهای تکراری

معرفی

متدهای حل مستقیم معادلات یک مشخصه مشترک و عمومی دارند که آن انجام محاسبات به وسیله تعداد محدودی از عملیات ریاضی است. به علاوه، اگر کامپیوترها قادر باشند که دقت بینهایت داشته باشند (بدون خطای گرد کردن)، حل دقیق خواهد بود.

متدهای تکراری یا غیر مستقیم، با یک حدس اولیه از حدس x آغاز میشوند و سپس به طور تکراری حل بهبود پیدا میکند تا زمانی که تغییرات در جواب x قابل چشم پوشی باشد یا اینکه به مقدار خطای تعیین شده برسیم. چون تعداد تکرارهای مورد نیاز میتواند خیلی زیاد باشد، متدهای غیر مستقیم، به طور کلی، کندتر از متدهای مستقیم هستند. هرچند که متدهای تکراری دارای مزایایی هستند که آنها را برای مساله های خاص جذاب کرده است، که این مزایا در زیر آمده است:گ

۱- در این متدها این امکان پذیر است که فقط المان های غیر صفر ماتریس ضرایب را نگه داریم. این قضیه این امکان را به ما میدهد که با ماتریس های بسیار بزرگ با مقادیر پراکنده کار کنیم، ماتریس هایی که محدود شده نیستند. در بسیاری از مسائل، نیازی نیست که اصلا ماتریس ضرایب را ذخیره کرده و نگه داریم.
۲- روش های تکراری روش های خود اصلاح هستند، به این معنا که خطاهای گرد کردن (یا اشتباهات محاسباتی) در یک حلقه تکرار در زیر حلقه ها تصحیح میشوند.

یکی از عیب های جدی متدهای مبتنی بر تکرار این است که این حل ها همیشه به مقدار درست حل همگرا نمیشوند. این بعدا نشان داده خواهد شد که فقط وقتی همگرایی تضمین میشود که ماتریس ضرایب از لحاظ قطری غالب باشد (غالبیت قطری). حدس اولیه x در این قضیه که همگرایی اتفاق بیفتد یا خیر، هیچ نقشی بازی نمیکند. اگر برای یک بردار آغازین متد همگرا شود، برای هر بردار شروعی این متد همگرا خواهد بود. حدس اولیه تنها بر روی تعداد تکرار مورد نیاز برای همگرایی موثر است.

متد گوس-سایدل

معادله $Ax = b$ در حالت اسکالر به صورت زیر نشان داده میشود:

$$\sum_{j=1}^{n} A_{ij}x_j = b_i, \quad i = 1, 2, \ldots, n$$

با خارج کردن ترم شامل x_i از علامت جمع، داریم:

$$A_{ii}x_i + \sum_{\substack{j=1 \\ j \neq i}}^{n} A_{ij}x_j = b_i, \quad i = 1, 2, \ldots, n$$

novin-eng.ir

www.novin-eng.ir

با حل کردن برای x_i به معادله زیر میرسیم:

$$x_i = \frac{1}{A_{ii}} \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} A_{ij} x_j \right), \quad i = 1, 2, \ldots, n$$

معادله آخری اسکیم تکراری زیر را پیشنهاد میدهد:

(۱)

$$x_i \leftarrow \frac{1}{A_{ii}} \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} A_{ij} x_j \right), \quad i = 1, 2, \ldots, n$$

ما با انتخاب بردار آغازین x شروع میکنیم. اگر یک حدس خوب برای حل در دسترس نباشد، x میتواند به صورت راندومی انتخاب گردد. معادله ۱ سپس برای محاسبه دوباره هر المان از x مورد استفاده قرار میگیرد، که همیشه از آخرین مقادیر در دسترس x_j استفاده میکند. این عملیات یک چرخه تکرار را کامل میکند. این روش تکرار میشود تا اینکه تغییرات x بین دو تکرار متوالی به انداز کافی کوچک شود.

همگرایی متد گوس سایدل به وسیله تکنیکی به نام "ضرایب استراحت" (relaxation factors) بهبود می یابد. ایده این روش بهبود در این است که مقدار جدید x_i را به صورت یک ضری وزنی از مقدار قبلی آن و مقدار پیش بینی شده توسط معادله ۱ قرار دهیم. فرمول متناظر با این قضیه به صورت زیر است:

(۲)

$$x_i \leftarrow \frac{\omega}{A_{ii}} \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} A_{ij} x_j \right) + (1 - \omega) x_i, \quad i = 1, 2, \ldots, n$$

که در اینجا ضری وزنی ω، ضریب استراحت نامیده میشود. همانطور که مشاهده میکنید اگر $\omega = 1$ باشد، هیچ اثر استراحتی مشاهده نمیشود، چون معادلات ۱ و ۲ باهم یکسان میشوند و جواب های یکسانی دارند. اگر $\omega < 1$ باشد معادله ۲ یک درونیابی را بین مقدار x_i قدیمی و جواب جدید معادله ۱ ایجاد میکند، این حالت زیر استراحت یا underrelaxation نامیده میشود. در کیس هایی که $\omega > 1$ ما یک برونیابی یا فوق استراحت (overrelaxation) داریم.

هیچ متد عملی و کاربردی برای تعیین کردن مقدار بهینه ω وجود ندارد. هرچند که، یک تقریب خوبی در زمان حل میتواند محاسبه گردد. فرض کنید $\Delta x^{(k)} = \left| x^{(k-1)} - x^{(k)} \right|$ مقدار تغیرات x در طول تکرار k ام باشد (بدون ضریب استراحت بدست آمده است یا با $\omega = 1$). اگر k به اندازه کافی بزرگ باشد ($k \geq 5$)، میتوان نشان داد که یک تقریب از مقدار بهینه ω به صورت زیر است:

(۳)

$$\omega_{\text{opt}} \approx \frac{2}{1+\sqrt{1-\left(\Delta x^{(k+p)}/\Delta x^{(k)}\right)^{1/p}}}$$

که در اینجا p یک عدد صحیح مثبت است.

موارد ضروری برای متد گوس-سایدل، وقتی که از ضرایب استراحت استفاده میکنید، به صورت زیر است:

۱- انجام دادن تعداد k تکرار با $\omega = 1$ (k=10) یک مقدار منطقی است. پس از تکرار k ام مقدار $\Delta x^{(k)}$ را ذخیره کنید.

۲- یک تکرار اضافی p ($p \geq 1$) را انجام دهید و سپس $\Delta x^{(k+p)}$ پس از آخرین تکرار ذخیره کنید.

۳- همه تکرارهای متعاقب را با $\omega = \omega_{opt}$ انجام دهید، که ω_{opt} از معادله ۳ بدست می آید.

کد متلب متد گوس سایدل

تابع gaussSeidel اعمال متد گوس سایدل به همراه ضرایب استراحت است. این کد به صورت اتوماتیک ω_{opt} را از معادله ۳ با استفاده از k=10 و p=1، محاسبه میکند. کاربر باید تابع iterEqs را مهیا کند که از فرمول ۲، مقدار x بهبود یافته را محاسبه میکند.

```
function [x,numIter,omega] = gaussSeidel(func,x,maxIter,epsilon)
% Solves Ax = b by Gauss-Seidel method with relaxation.
% USAGE: [x,numIter,omega] = gaussSeidel(func,x,maxIter,epsilon)
% INPUT:
% func = handle of function that returns improved x using
% the iterative formulas in Eq. (2).
% x = starting solution vector
% maxIter = allowable number of iterations (default is 500)
% epsilon = error tolerance (default is 1.0e-9)
% OUTPUT:
% x = solution vector
```

novin-eng.ir