تمرین سری هشتم درس روش‌های عددی بهینه‌سازی

امیرحسین افشارراد
۹۵۱۰۱۰۷۷

۲۵ دی ۱۳۹۸

۱ مروری بر نتایج تمرین هفتم

آن چه در ادامه‌ی این صفحه مشاهده می‌کنید، دقیقاً همان گزارشی است که برای تمرین سری هفتم (پیاده‌سازی الگوریتم‌های نیوتن و SD برای دو تابع داده‌شده به کمک line search با شرایط ولف) تحویل داده شد. در ادامه نتایج به دست آمده در تمرین جدید را با این نتایج مقایسه می‌کنیم.

نکته ۱. دو تابع برای پیاده‌سازی Line Search نوشته شده است؛ با نام‌های linesearch و zoom که به ترتیب مربوط به مراحل bracketing و sectioning هستند و بدنه‌ی اصلی کد آن‌ها از کتاب نوسدال گرفته شده است و برخی جزئیات به آن اضافه شده است.

نکته ۲. سه پارامتر مهم مسأله به صورت $c_1 = 10^{-4}$ و $c_2 = 0.1$ و $\bar{f} = 0.001$ انتخاب شده‌اند. c_1 و c_2 ضرایب موجود در شروط اوّل و دوم ولف هستند و $\bar{f}$، ثابتی است که (مطابق با توضیحات جزوه‌ی درس) برای تعیین $\alpha_{\max}$ و نیز اتمام زودهنگام فرایند linesearch (در صورت لزوم) به کار می‌رود.

نکته ۳. در آغاز کد مربوط به تابع zoom، نیاز به انجام یک درون‌یابی داریم. این درون‌یابی را با یک تابع درجه دوم انجام داده‌ایم و در ادامه مطابق با توضیحات جزوه، مقدار α_j جدید را از مقایسه‌ی محلّ مینیمم این سهمی با نقاط ابتدا و انتهای بازه به دست آورده‌ایم.

نکته ۴. مطابق با توضیحات جزوه‌ی درس، برای حالتی که $|\phi'(a_j)|$ از مقداری مانند ϵ کوچک‌تر می‌شود، الگوریتم sectioning متوقّف می‌شود و پیامی نیز بر روی صفحه چاپ می‌شود که همین موضوع را اطّلاع‌رسانی می‌کند. البته لازم به ذکر است در حلّ دو مسأله‌ی بهینه‌سازی مربوط به این تمرین، چنین موردی پیش نمی‌آید. همچنین مقدار ϵ مذکور برابر با 10^{-10} انتخاب شده است.

نکته ۵. مقدار α_1 در تابع linesearch برابر با ۱ در نظر گرفته شده است. این یکی از روش‌هایی است که در جزوه پیشنهاد شده است، و همچنین در جزوه گفته شده که این روش برای الگوریتم نیوتن مناسب‌تر است؛ و جالب است که نهایتاً در نتایج نیز می‌بینیم که عملکرد الگوریتم نیوتن بهتر است (و شاید دلیل آن مربوط به این موضوع باشد).

در ادامه نتایج را مشاهده می‌کنید:

	final $\mathbf{x}$	final f	number of iterations	number of function evaluations	number of gradient evaluations	number of Hessian evaluations
Rosenbrock function	$\begin{bmatrix}1.3507\\1.8255\end{bmatrix}$	0.1231	3	120	54	0
Powel function	$\begin{bmatrix}0.2389\\-0.0238\\0.1139\\0.1237\end{bmatrix}$	0.0062	213	13266	8051	0

Table 1: Steepest Descent Method

	final $\mathbf{x}$	final f	number of iterations	number of function evaluations	number of gradient evaluations	number of Hessian evaluations
Rosenbrock function	$\begin{bmatrix} 1.0000 \\ 1.0000 \end{bmatrix}$	0	2	5	4	2
Powel function	$\begin{bmatrix} 0.0000 \\ -0.0000 \\ 0.0011 \\ 0.0011 \end{bmatrix}$	3.5879×10^{-11}	12	35	33	12

Table 2: Newton Method

۲ نتایج الگوریتم BFGS (Line Search) با شرایط وُلف)

حال نتایج را با آن چه در صفحات قبل مشاهده می‌شود (مربوط به تمرین سری هفتم) مقایسه می‌کنیم:

- تابعی به نام BFGS نوشته شده است که با دریافت تابع، گرادیان تابع، نقطه‌ی شروع، و شرط توقّف (پارامتری که هرگاه نُرم ∇f از آن کم‌تر شد، الگوریتم متوقّف شود)، الگوریتم BFGS را پیاده‌سازی می‌کند.
- برای محاسبه‌ی C_0 (مقدار اوّلیه‌ی ماتریس C که خود، تخمینی از وارون ماتریس هسیَن است)، مطابق با روشی که در جزوه گفته شده است، ابتدا یک بار مقدار آن را برابر با ماتریس همانی در نظر می‌گیریم و یک iteration الگوریتم را اجرا می‌کنیم تا مقادیر $\delta_0 = x_1 - x_0$ و $\gamma_0 = \nabla f(x_1) - \nabla f(x_0)$ محاسبه شوند، سپس مقدار اوّلیه‌ی C را به صورت زیر قرار داده و الگوریتم را از ابتدا اجرا می‌کنیم:

$$C_0 = \frac{\gamma_0^T \delta_0}{\delta_0^T \delta_0} I$$

در ادامه نتایج اجرای این الگوریتم را برای دو تابع روزنبروک و پاول مشاهده می‌کنید:

	final $\mathbf{x}$	final f	number of iterations	number of function evaluations	number of gradient evaluations	number of Hessian evaluations
Rosenbrock function	$\begin{bmatrix}0.9999\\0.9999\end{bmatrix}$	5.09×10^{-9}	12	230	123	0
Powel function	$\begin{bmatrix}-0.0074\\0.0007\\0.0028\\0.0028\end{bmatrix}$	1.1×10^{-7}	20	327	173	0

Table 3: BFGS Method

مشاهده می‌شود که در ازای محاسبه نکردن وارون ماتریس هسین، مجبور شده‌ایم تعداد دفعات محاسبه‌ی تابع اصلی و گرادیان آن را (در مقایسه با روش نیوتن عادی) افزایش دهیم؛ امّا از نظر دقّت و کیفیت نتایج چیزی را از دست نداده‌ایم و پاسخ الگوریتم بسیار عالی است.