Bitácora 14

Cálculo e Análise Numérica

19 de marzo 2024

59 asistentes

Laura Becerro Haro
Marina Collazo Couñago
Atenea García González
Sergio Souto Mourelle

Índice

1. Corrección da bitácora anterior

A clase comeza coa revisión e corrección da Bitácora 13, sobre a cal a docente, malia comentar que está bastante ben feita, resalta certos erros.

Canto a corrección da bitácora 12 menciónanse dúas pequenas grallas: a falta dun punto final tras escribir o Hessiano dunha función f no punto (x_0, y_0) e un erro tipográfico na palabra "*Atopar*".

Por outra banda, xa entrando na explicación do contido da clase do luns, acháronse algunhas inexactitudes. No método do descenso rápido definiuse G como a lonxitude do vector $f(x_1, x_2)$, cando en realidade é a lonxitude do vector $f(x_1, x_2)$ ao cadrado. Outro erro salientado é o uso de parénteses extra na definición de $-\nabla G(x_1, x_2)$, mais mencionouse que probablemente aconteceu porque esta errata tamén aparecía no *PDF* do Campus Virtual, o cal se correxirá pronto. Ademais do xa mencionado, tamén comentouse a falta de orde na redacción da explicación da definición dos iterantes do método do descenso rápido, dificultando a súa comprensión. Finalmente, a gráfica proposta para resumir o procedemento do método do descenso rápido contradí o explicado na propia bitácora. Proponse a gráfica 1, da práctica de extremos, como alternativa.

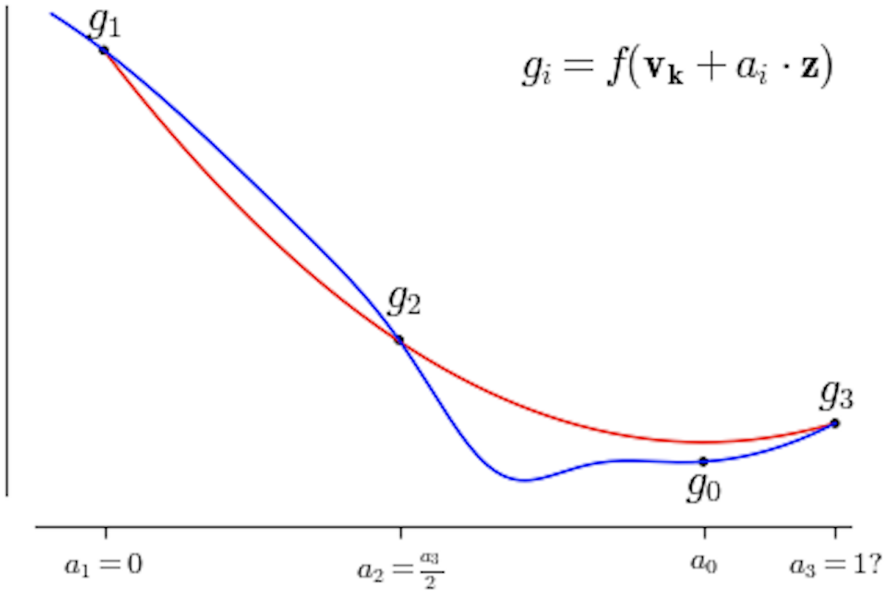

Figura 1: Gráfico dos iterantes do método de descenso rápido.

1.1. Explicación tarefa 2

Con respecto á tarefa 2, a docente aproveita a explicación feita na bitácora para matizar e falar con detalle de certos puntos do traballo. Nela afirmábase que a entrega debía facerse na semana do 22 de abril, pero en realidade nesas datas realizarase a exposición oral da tarefa. Será despois disto cando se entregue, permitindo así a corrección de posibles erros atopados durante a presentación. Cabe destacar que as presentacións faranse no horario de prácticas e na aula habitual de Monte da Condesa.

Debido a que a tarefa pode realizarse individualmente ou por parellas (e recoméndase esta segunda opción, xa que favorece a adquisición de competencias baseadas no traballo en equipo), ambas persoas deberán decidir que día realizar a exposición. Preferiblemente os dous membros da parella pertencerán ao mesmo grupo de prácticas, pero en caso de non ser así terán que escoller

un día no que ambos estén dispoñibles. Para isto habilitouse no Campus Virtual un apartado onde facer a escolla do día no que se quere realizar a presentación, así como para indicar que persoas conforman cada grupo, do mesmo xeito que se fixera ao comezo do curso coas bitácoras. Existe a posibilidade de escoller un horario fóra do das prácticas, pero para isto débese ter una causa xustificada que impida acudir a calquera outra sesión. Nesta mesma entrada do Campus Virtual especifícanse tamén as instrucións detalladas para a realización da tarefa.

A puntuación desta tarefa é de 0,6 puntos coma máximo, os cales non poderán ser recuperados en ningún outro momento do curso, xa que a avaliación continua remata co período de clases. A docente resaltou que antes había unha gratificación de 0,5 puntos por facer o traballo en parella e ter bos resultados, pero, tras o cambio no porcentaxe de puntuación asignada á avaliación continua (pasou de valer un 30 % a un 40 % da nota) isto deixou de facerse. Esta aclaración fíxose para evitar malentendidos no caso de que algún ex-alumno o dixera.

Pasando a falar das prácticas que se realizarán ao longo da semana, xa están subidas aquelas relacionadas tanto co método de Newton para sistemas coma co método do descenso rápido. Estas prácticas servirán para comprender e asimilar os conceptos e código precisos para realizar a tarefa 2. Aproveitando isto, ensínanse algunhas das gráficas da práctica de extremos para comentar o que se ensinou durante a clase.

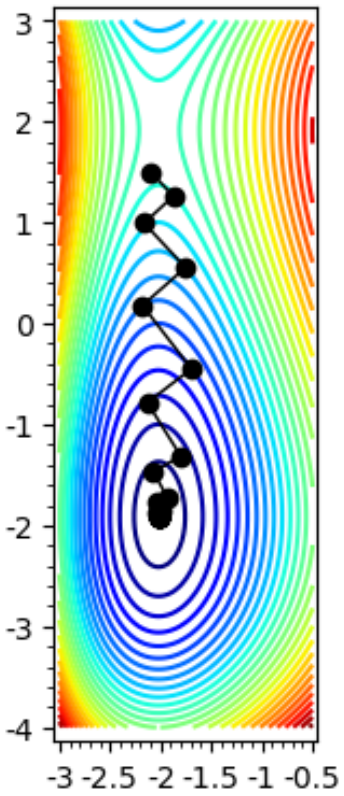

Figura 2: Gráfica dos iterantes do método do descenso rápido

2. Explicación do método de descenso rápido

Retomando agora o visto na clase de teoría do luns, aclarouse a explicación do método de descenso rápido.

O problema que resolve o método de descenso rápido, ao igual que o método de Newton, é a resolución de sistemas non lineais.

PROBLEMA 1: Sexa $\boldsymbol{f} : Df \subseteq \mathbb{R}^2 \to \mathbb{R}^2$ (ou, xeralizando, $\boldsymbol{f} : Df \subseteq \mathbb{R}^n \to \mathbb{R}^n$), achar $\boldsymbol{\alpha}$ tal que $\boldsymbol{f}(\boldsymbol{\alpha}) = \boldsymbol{0}$.

A partir do *PROBLEMA 1* pódese formular o seguinte problema, que terá a mesma solución: Defínese como función a minimizar

$$G(x_1, x_2) := \|\boldsymbol{f}(x_1, x_2)\|^2 = (f_1(x_1, x_2))^2 + (f_2(x_1, x_2))^2.$$

Polo que G representa o cadrado da lonxitude do vector $f(x_1, x_2)$. Ademais, como $\mathrm{Im}G \subseteq [0, +\infty)$, $\boldsymbol{\alpha}$ coincidirá co valor no que G acada un mínimo absoluto.

Para minimizar G a partir dun punto dado $\boldsymbol{X}_0 = (x_1^0, x_2^0)$ seguirase a dirección de $-\nabla G(x_1, x_2)$, xa que, partindo das propiedades do vector gradiente, sempre apuntará á dirección de máximo decrecemento de G partindo de $\boldsymbol{X}_0$.

2.1. Cálculo de $\nabla G(x_1, x_2)$

Podemos redefinir a función G como a composición seguinte:

$$(x_1, x_2) \rightsquigarrow (f_1(x_1, x_2) f_2(x_1, x_2)) \rightsquigarrow z_1^2 + z_2^2 = \|\boldsymbol{f}(x_1, x_2)\|^2 = G(x_1, x_2).$$

Así que $\nabla G(x_1, x_2)$ será:

$$\nabla G(x_1, x_2) = 2(f_1(x_1, x_2) f_2(x_1, x_2)) Df(x_1, x_2) =$$
$$(2f_1(x_1, x_2) f_1(x_1, x_2)_{x_1} + 2f_2(x_1, x_2) f_2(x_1, x_2)_{x_1} \quad 2f_1(x_1, x_2) f_1(x_1, x_2)_{x_2} + 2f_2(x_1, x_2) f_2(x_1, x_2)_{x_2}).$$

2.2. Definición dos iterantes do método

Agora que coñecemos a dirección de máximo decrecemento, necesitamos coñecer "canto" hai que moverse nela.

Noutras palabras, dada a ecuación da recta que pasa polo iterante inicial $\boldsymbol{X}_0 = (x_1^0, x_2^0)$ na dirección unitaria de máximo decrecemento $-\nabla G(x_1^0, x_2^0)/\|\nabla G(x_1^0, x_2^0)\|$:

$$\boldsymbol{X}_1 := \boldsymbol{X}_0 - a\nabla G(\boldsymbol{X}_0)/\|\nabla G(\boldsymbol{X}_0)\|$$

haberá que achar a.

2.2.1. Cálculo do valor de a

PROBLEMA 2: Achar o valor de a que minimiza a función G sobre a recta definida no parágrafo anterior. É dicir, achar o valor de a que minimiza a función:

$$h(a) := G(\boldsymbol{X}_0 - a\nabla G(\boldsymbol{X}_0)/\|\nabla G(\boldsymbol{X}_0)\|).$$

Os pasos a seguir para resolver o *PROBLEMA 2* son:

1. Comézase no punto $\boldsymbol{X}_0$, que corresponde coa elección $a = 0$, que notamos por a_1. Chamaremos g_1 ao valor de G correspondente

$$g_1 = G(\boldsymbol{X}_0 - a_1 \nabla G(\boldsymbol{X}_0)/\|\nabla G(\boldsymbol{X}_0)\|).$$

2. Agora tomaremos a elección $a = 1$, que denotamos por a_3. Chamaremos g_3 ao valor de G correspondente

$$g_3 = G(\boldsymbol{X}_0 - a_3 \nabla G(\boldsymbol{X}_0)/\|\nabla G(\boldsymbol{X}_0)\|).$$

   Para valorar se esta elección de $a3$ é boa, terase en conta o seguinte:

   a) Se $g_3 > g_1$ pasouse de largo o mínimo de h. Polo tanto, redefinimos a_3 como a metade do anterior ata que $g_3 < g_1$.
   Se o valor de a_3 atopado cando se consiga cumplir $g_3 < g_1$ é menor que a metade da tolerancia, seguimos moi preto de $\boldsymbol{X}_0$ e non hai mellora do iterante, así que rematamos o proceso.

   b) Se $g_3 < g_1$, a elección de a_3 é correcta.

3. Neste momento, xa se asegurou que $g_3 < g_1$ e que a_3 cumple o criterio da tolerancia. Agora redefinimos $a_2 := a_3/2$.

   - Para obter un valor ópimo de a construirase o polinomio de interpolación de grao dous da función h de xeito que pase polos puntos a_1, a_2 e a_3:

$$P(a) = g_1 + h_1 a + h_3(a - a_1)(a - a_2),$$

     onde

$$h_1 = (g_2 - g_1)/a_2, \quad h_2 = (g_3 - g_2)/(a_3 - a_2), \quad h_3 = (h_2 - h_1)/a_3.$$

   - Calcúlase o punto a_0 onde P ten un mínimo:

$$a_0 = (a_2 - h_1/h_3)/2,$$

     e o seu valor correspondente de G, que chamaremos g_0:

$$g_0 = G(\boldsymbol{X}_0 - a_0 \nabla G(\boldsymbol{X}_0)/\|\nabla G(\boldsymbol{X}_0)\|).$$

4. Para finalizar, hai agora 4 posibles candidatos a mínimo (a_0, a_1, a_2 e a_3). Pola elección feita poderíamos descartar a_1. De entre os restantes, seleccionarase aquel que proporcione o menor valor de G.

O valor escollido para a definirá o novo iterante $\boldsymbol{X}_1$. Polo tanto, a sucesión do método do descenso rápido ven dada por:

$$\boldsymbol{X}_{n+1} := \boldsymbol{X}_n - a \nabla G(\boldsymbol{X}_0)/\|\nabla G(\boldsymbol{X}_0)\|,$$

onde o valor de a calcúlase en cada iteración seguindo o procedemento descrito antes.

3. Sistemas lineais I

Falando agora do novo temario, introdúcense os sistemas de ecuacións lineais. Para dita explicación, a docente apoiase nos apuntes da Universidade de Concepción, situada en Chile. Aproveita para salientar que, aínda que puidera parecer máis natural explicar antes os sistemas de ecuacións lineais en lugar dos non lineais, a orde que se está a tomar permítenos reforzar conceptos aprendidos no tema anterior, coma o vector gradiente ou a regra da cadea.
Tamén comenta a importancia da adquisición de coñecementos relacionados cos sistemas lineais para o grao que estamos a cursar, xa que para as empresas é de vital importancia que os enxeñeiros informáticos sexan quen de implementar algoritmos que resolvan este tipo de problemas de maneira rápida e eficaz.

O obxectivo deste tema é comprender os sistemas de ecuacións lineais e os métodos existentes para resolvelos, así como aprender a distinguir que métodos son máis rápidos para a resolución de sistemas de grande magnitude.

3.1. Representación matricial

Todos os sistemas de ecuacións lineais pódense expresar como **matrices**:

$$\begin{cases} a_{11}x_1 + ... + a_{1n}x_n = b_1 \\ \vdots \qquad\qquad \vdots \\ a_{n1}x_1 + ... + a_{nn}x_n = b_n \end{cases} \iff \mathbf{Ax} = \mathbf{b}$$

onde

$$A := \begin{pmatrix} a_{11} & ... & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & ... & a_{nn} \end{pmatrix} \in \mathbb{R}^{nxn} \qquad \text{e} \qquad b = \begin{pmatrix} b_1 \\ \vdots \\ a_n \end{pmatrix} \in \mathbb{R}^{nxn}$$

son os datos e $x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^{nxn}$ é o vector de incógnitas.

Sempre consideraremos matrices **A** cadradas.

3.2. Matriz inversa

O sistema de ecuacións lineais $\mathbf{Ax} = \mathbf{B}$ ten solución única se, e só se, a matriz é *non singular*. Unha matriz $A \in \mathbb{R}^{nxn}$ é non singular se, e só se, cumpre calquera destas condicións:

- **A** é **invertible**: $\exists \mathbf{A}^{-1} \in \mathbb{R}^{nxn}$ tal que $\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = I$
- $|\mathbf{A}| \neq 0$.
- Todas as filas e columnas de **A** son linealmente independentes: rango$(\mathbf{A}) = n$
- 0 non é un valor propio de **A**: $0 \notin \sigma(\mathbf{A})$

Se $\mathbf{A}$ é non singular, entón $\mathbf{Ax} = \mathbf{B} \rightarrow \mathbf{x} = \mathbf{A^{-1}B}$. Porén, en xeral, calcular $\mathbf{A}^{-1}$ é moi costoso computacionalmente, polo que non é conveniente resolver o sistema deste xeito.

No seu lugar, outra maneira de calcular a inversa dunha matriz $\mathbf{A} \in \mathbb{R}^{nxn}$ consiste en resolver n sistemas de ecuacións lineais. Chamando $c^1, ..., c^n$ ás columnas de $\mathbf{A}^{-1}$, temos que:

$$A^{-1} := \left[\begin{array}{c|c|c} c_1 & ... & c_n \end{array}\right], \, c^1, ..., c^n \in \mathbb{R}^n.$$

Entón:

$$\left[\begin{array}{c|c|c} Ac^1 & ... & Ac^n \end{array}\right] = \mathrm{A} \left[\begin{array}{c|c|c} c_1 & ... & c_n \end{array}\right] = AA^{-1} = \mathrm{I} = \left[\begin{array}{c|c|c} e_1 & ... & e_n \end{array}\right],$$

onde as columnas $e^1, ..., e^n$ de $\mathbf{I}$ son os vectores da base canónica de $\mathbb{R}^n$. Isto permítenos calcular $\mathbf{A}^{-1}$ columna por columna resolvendo:

$$Ac^i = e^i, \qquad i = 1, ..., n$$

Mentres se estaba a explicar isto, plantexouse a dúbida de se existe algunha diferencia entre unha matriz *non singular* e unha matriz *regular*. A profesora resolverá esta cuestión na seguinte clase expositiva.

3.3. Dificultades numéricas

Ao deseñar un algoritmo co obxectivo de resolver un sistema de ecuacións lineais, hai que ter en conta o custo operacional, o custo de almacenaxe e a precisión dos resultados.

O **custo operacional** refírese ao tempo de cálculo que precisa o computador para resolver o sistema. Débese intentar reducir este tempo ao mínimo. O **flop** ***(floating point operations)*** é a unidade estándar para medir este custo.

O **custo de almacenaxe** refírese a cantidade de posicións de memoria requiridas para a execución do algoritmo, a cal tamén se deben minimizar tanto como sexa posible. Estas representan os datos gardados, as variables auxiliares e demais.

A **precisión dos resultados** é de gran importancia, xa que, se o algoritmo non é estábel, os errores de datos e de redondeo pódense amplificar e resultar nunha saída errónea.

A profesora destacou tamén a diferencia entre os problemas reais e os problemas "triviais" coma os que resolvemos a man na clase, xa que nos segundos o truncamento ou o redondeo de decimais e practicamente inconsecuente, mais nos problemas reais con milleiros de operacións un cambio nun só decimal pode supoñer unha diferencia moi grande nos resultados.

3.3.1. Custo operacional

Na maioría de aplicacións actuais, os sistemas son de gran tamaño. Un sistema de 1000x1000 considerase hoxe en día coma un sistema doado. De feito, nalgunhas aplicacións téñense que resolver sistemas de centos de milleiros de incógnitas. Para a súa resolución existen métodos que permiten

solventar, en principio, calquer tipo de sistema, pero, a efectos prácticos, o tempo requirido sería masivo.

Un destes métodos é a **Regra de Cramer**, que trata de calcular a solución do sistema $\boldsymbol{Ax} = \boldsymbol{b}$ mediante:

$$x_i = \frac{\det(\boldsymbol{A_i})}{\det(\boldsymbol{A})}, \qquad i = 1, ..., n$$

onde $\boldsymbol{A_i}$ é a matriz obtida a partir da substitución da columna i-ésima de A polo segundo membro do sistema, b.

O problema é que o custo operacional deste método é de $(n+1)!$ flops.

En cambio, o **Método de Eliminación Gaussiana** baséase no método alxébrico das *transformacións elementais*, alcanzando un baixo custo operacional de aproximadamente $\frac{2}{3}n^3$ flops.

Na figura 3 ilústrase a comparación dos tempos de cómputo de sistemas de distintos tamaños usando estes dous métodos.

En un computador de 1 Gflop (10^9 flop) por segundo:

n	10	15	20	100	1000	2000
Regla de Cramer						
flop	4×10^7	2×10^{13}	5×10^{19}	10^{160}	"∞"	"∞"
tiempo	0.04 s	5.5 horas	1500 años	"∞"	"∞"	"∞"
Eliminación Gaussiana						
flop	666	2250	5333	7×10^5	7×10^8	5×10^9
tiempo	0. s	0. s	0. s	0 s	0.73 s	4.88 s

Figura 3: Gráfico comparativo da Regra de Cramer e a Eliminación Gaussiana

É importante destacar tamén a relevancia do gasto enerxético que conlevan os métodos con elevados custos operacionais.

3.3.2. Custo de almacenaxe

Existe un gran número de matrices nas aplicacións de sistemas de ecuacións lineais que están compostas principalmente por entradas nulas. Este tipo de matrices denomínanse *dispersas* ou *ralas* e existen métodos que aproveitan esta situación para optimizar o espazo de almacenaxe, ocupando aproximadamente tantas posicións de memoria coma entradas non nulas hai na matriz.

Os métodos alxébricos máis comúns, coma o método de *transformacións elementais*, modifican a matriz orixinal e, por consecuencia, desfanse da calidade dispersa desta. En cambio, outros procedementos, os *métodos iterativos*, non modifican a matriz do sistema, polo que son máis óptimos en canto ao custo de almacenaxe, xa que en lugar de calcular a solución exacta do sistema, crean iterativamente aproximacións cada vez mellores.

3.3.3. Precisión dos resultados

A propagación de erros nos datos e no redondeo é inevitable, polo que é necesario dispoñer de técnicas que permitan **predicir** cando estes erros propagaranse drásticamente na resolución dun sistema.

Ademais, é necesario tamén deseñar métodos numéricos **estables** que reduzan a propagación de erros de redondeo a un mínimo e **técnicas computacionais** que permitan estimar a precisión dunha solución despois de calculala para comprobar se o erro é menor que a tolerancia aceptable.

En programas meteorolóxicos coma MeteoGalicia a precisión é de extrema importancia, xa que a gran cantidade de operacións necesarias para predicir correctamente o tempo e a magnitude das cifras tratadas fai que un erro poda levar a resultados moi afastados da realidade.

Ao final da clase a profesora mencionou o método de Cholesky, o cal optimiza a resolución de sistemas asociados a matrices simétricas, e menciona que na seguinte clase se falará da factorización LU.