دانشگاه صنعتی شریف
دانشکده علوم ریاضی

پایان‌نامه کارشناسی ارشد
ریاضی کاربردی

عنوان

حذف گره در منحنیهای B-اسپلاین

استاد راهنما

دکتر نظام‌الدین مهدوی امیری

نگارش

سیّد علی کتان‌فروش

مهر ۱۳۷۸

قدردانی

حذف گره در منحنیهای B-اسپلاین

چکیده

هدف از حذف گره در یک منحنی B-اسپلاین، بدست آوردن یک منحنی B-اسپلاین جدید، تعریف شده بر روی دنباله گره‌های حاصل از حذف برخی گره‌ها از یک دنبالهٔ اولیه است بطوریکه دارای کمترین اختلاف با منحنی اولیه بر حسب برخی نرمها باشد.

در این رساله مسألهٔ حذف یک گره درونی از دنبالهٔ گره‌های یک منحنی B-اسپلاین مورد بررسی قرار می‌گیرد. برای این کار ابتدا یک ساختار هندسی موضعی برای نقاط کنترلی جدید مبتنی بر دو معکوس خاص ماتریس درج گره ارائه می‌شود. سپس ضرایب وزنی آزادی که در این ساختار کلی ظاهر می‌شوند، بوسیلهٔ مینیمم‌سازی خطای بین دو منحنی اولیه و جدید بر اساس سه نرم L_∞ ، L_2 گسسته و L_∞ پیوسته، بدست می‌آیند. متناظر با هر یک از این نرمها یک مسألهٔ تقریب مطرح می‌شود که بهترین آنها، تقریب L_∞ پیوسته بوسیلهٔ یک نوع خاص از الگوریتم رِمز محاسبه می‌گردد.

با پیاده‌سازی چند الگوریتم ترکیب و تجزیه B-اسپلاین، مبتنی بر روشهای حذف تک گره، برخی کاربردها مانند مسألهٔ رتبه‌بندی گره‌ها در حذف همزمان چند گره بعنوان یک کاربرد خاص و چندین کاربرد عملی مثل برازش منحنی و رویه، پردازش امواج، فشرده‌سازی تصاویر و تقریب توابع روی فضای اسپلاینها با دنباله گره‌های آزاد را می‌توان مورد بررسی قرار داد.

واژه‌های کلیدی: حذف گره، درج گره، تقریب L_∞ گسسته، تقریب L_2 گسسته، تقریب L_∞ پیوسته، الگوریتم رمز، ترکیب و تجزیهٔ B-اسپلاینها، دنبالهٔ گره‌های غیریکنواخت، رتبه‌بندی گره‌ها، پردازش امواج، مسألهٔ بهترین گره‌ها.

پیشگفتار

طراحی هندسی یکی از زمینه‌های علوم کامپیوتر است که از چند دههٔ پیش تاکنون بطور گسترده‌ای در زمینه‌های طراحی صنعتی، هنر گرافیک و پویانمایی و بسیاری از مسائل فیزیک و مهندسی بکار می‌رود. طراحی هندسی را می‌توان زاییده موضوعاتی چون آنالیز عددی و نظریه تقریب در ریاضیات و طراحی و مدلسازی گرافیکی در کامپیوتر دانست که برای پاسخگویی به نیازهای روزافزون صنعت و مهندسی بوجود آمد. با این حال این شاخهٔ پرکاربرد تاکنون چندان در بین ریاضیدانان و ریاضیکاران کشورمان که در گرایشهای کاربردی فعالیت می‌کنند بطور فراگیر مورد توجه قرار نگرفته است.

کاربردهای اولیهٔ طراحی هندسی عمدتاً در زمینهٔ انتخاب و سپس بکارگیری برخی مدلهای تعبیرپذیر برای بیان پدیده‌های هندسی و انعطاف پذیر برای کنترل خواص هندسی آنها بود. Bـاسپلاین یکی از بنیادی‌ترین مدلها در بین آنهاست. با توسعه این مدلها و تحقیقات بر روی برخی قابلیتهای جدید در آنها، دامنهٔ کاربردهایشان باندازهٔ زیادی گسترش یافت. از جملهٔ این کاربردهای نسبتاً جدید مبحث روشهای ترکیب و تجزیه است که کاربردهای آن از مسائل فیزیک و پردازش امواج تا فشرده‌سازی داده‌ها و تصاویر در کامپیوتر، گسترده شده است. این روشها عمدتاً درگیر با مفهوم جدید موجک هستند. هم اکنون موجکها بعنوان یک مدل قوی و کارآمد در صدر مباحث طراحی هندسی بکار گرفته می‌شوند.

مسألهٔ حذف گره در Bـاسپلاینها بعنوان یک رهیافت طبیعی و ساده برای بررسی روشهای ترکیب و تجزیه در این میان جلوه می‌کند. روش حذف گره که گاهی بعنوان روش کاهش داده‌ها بررسی می‌شود، به بیان خیلی ساده، فرایندی است را که در آن سعی می‌شود یک منحنی یا رویه که با تعداد زیادی از نقاط داده مدل شده است، با یک منحنی یا رویهٔ دیگر با تعداد کمتری داده بنحوی تقریب زد که اختلاف بین آنها برحسب برخی نرمها کمترین مقدار ممکن باشد. این عمل از یک دیدگاه نقطهٔ مقابل درج گره قرار دارد. در

درج گره برعکس سعی می‌شود با افزودن تعداد داده‌های یک منحنی یا رویه، داده‌هایی بدست آورد که با ظرافت بیشتری منحنی یا رویه خود را مدلسازی کنند. در این رساله یک حالت مقدماتی حذف گره، بطور خاص مورد بررسی قرار می‌گیرد. در این حالت خاص به مسألهٔ حذف فقط یک گره از منحنی Bـاسپلاین از دیدگاه نظری و عملی پرداخته می‌شود و در انتها مجموعه‌ای از کاربردهای عملی آن به نمایش گذاشته می‌شود.

برای آشنایی با برخی مفاهیم مقدماتی دربارهٔ Bـاسپلاینها بطور فشرده مطالبی در فصل اول آمده است. در فصل دوم به موضوع درج گره در Bـاسپلاینها پرداخته می‌شود و ایده‌های اولیه برای حل مسأله حذف گره پی‌ریزی می‌گردد. در فصل سوم به موضوع اصلی رساله بطور مفصل و تحلیلی پرداخته می‌شود. در فصل چهارم، پیاده‌سازی عملی الگوریتمهای حذف گره تشریح می‌شود و در فصل پنجم فهرستی از کاربردهای این الگوریتمها مرور می‌شود. در ضمیمهٔ A، تمام الگوریتمهای تشریح شده در متن بصورت برنامه‌هایی به زبان MATLAB آورده شده‌اند.

در اینجا شایسته است از استادان محترم آقایان دکتر مهدوی امیری و دکتر مهری که عمدهٔ آموخته‌های خود را مدیون ایشان هستم و همچنین از آقای دکتر سماواتی که هدایتگر و زمینه‌ساز علاقه‌مندی نگارنده به گرایش طراحی هندسی هستند و نیز بخاطر مساعدت و راهنمایی‌های ارزندهٔ ایشان در تألیف این رساله، با نهایت امتنان، تشکر کنم. از شما خوانندهٔ محترم نیز که با مطالعهٔ موشکافانهٔ خود، اشکالات و معایب احتمالی این رساله را به اطلاع نگارنده می‌رسانید، پیشاپیش تشکر می‌کنم.

email: *katanfor@math.sharif.ac.ir*

فهرست

۱ Bـاسپلاینها و طراحی هندسی

۱—۱ معرفی ۱

۱—۲ پایه‌های Bـاسپلاین ۳

۱—۳ منحنی Bـاسپلاین ۷

۱—۳—۱ خواص منحنی Bـاسپلاین ۸

۱—۳—۲ ترکیب توابع پایهٔ Bـاسپلاین ۱۱

۱—۳—۳ مشتق ۱۲

۱—۳—۴ تولید منحنی‌های آزاد ۱۳

۱—۴ منحنی بزیه ۱۵

۱—۴—۱ خواص منحنی بزیه ۱۵

۱—۴—۲ تظریف منحنی بزیه ۱۷

۲ درج گره

۲–۱ کلیات ۲۰

۲–۲ درج تک گره ۲۱

۲–۲–۱ الگوریتم درج تک گره ۲۷

۲–۲–۲ درج مکرر یک گره ۲۸

۲–۳ درج چند گره ۳۰

۳ حذف گره

۳–۱ زمینه ۳۳

۳–۱–۱ کاهش درجه در منحنی بزیه ۳۵

۳–۱–۲ ضرایب براکتی ۳۶

۳–۲ حذف گره مجازی ۴۰

۳–۲–۱ ضریب خطای حذف گره ۴۳

۳–۳ حذف گره بهترین تقریب ۴۶

۳–۳–۱ تعریف یک تابع خطا ۴۷

۳–۳–۲ تقریب L_∞ گسسته ۵۱

۳–۳–۳ تقریب L_2 گسسته ۵۳

۳–۳–۴ تقریب L_∞ پیوسته ۵۷

۴ پیاده‌سازی

۴–۱ پیاده‌سازی الگوریتم رمز ۶۴

۴–۲ روشهای ترکیب و تجزیه ۷۴

۴–۳ نتایج ۸۲

۵ کاربردها

۵–۱ رتبه بندی گره‌ها برای الگوریتمهای حذف همزمان چند گره ۹۰

۵–۲ برازش و کاهش داده‌ها ۹۳

۵–۲–۱ برازش منحنی ۹۳

۵–۲–۲ پردازش امواج ۹۶

۵–۲–۳ فشرده‌سازی و پردازش تصاویر ۹۷

۵–۳ تقریب توابع و مسألهٔ بهترین گره‌ها ۱۰۲

A برنامه‌هایی به زبان MATLAB ۱۰۵

فصل ۱

Bـاسپلاینها و طراحی هندسی

۱ـ۱ معرفی

توابع Bـاسپلاین بعنوان یک ابزار کارا از گذشته در شاخه‌های گوناگون ریاضیات به خصوص در آنالیز عددی، نظریه تقریب و از چند دهه پیش تاکنون در طراحی هندسی مورد استفاده قرار گرفته‌اند. این توابع از این جهت که تشکیل یک پایه برای فضای اسپلاینها می‌دهند و چون اساساً بوسیله روشهای عددی پایدار قابل محاسبه‌اند و نیز بواسطهٔ تأثیر موضعیشان در تقریب توابع، به طور گسترده در مباحث مختلف آنالیز عددی از حل عددی معادلات دیفرانسیل و انتگرال و مسائل وردشی تا قضایای مجرد نظریه تقریب دیده می‌شوند.

هدف طراحی هندسی ارائهٔ مدلهای ریاضی مناسب برای نمایش پدیده‌های هندسی در دنیای فیزیکی مانند منحنیها، رویه‌ها و ... و بررسی و مطالعهٔ خواص چنین مدلهایی است. این شاخه در علوم کامپیوتر بعنوان یک پشتوانهٔ نظری برای طراحی به کمک کامپیوتر[۱] بکار می‌رود. طراحی هندسی به کمک کامپیوتر[۲] از این ارتباط بوجود می‌آید.

اینکه مدلهای ریاضی ارائه شده در طراحی هندسی دارای چه ویژگیهایی باشند در ابتدای امر چندان

[۱] Computer Aided Design: CAD

[۲] Computer Aided Geometric Design: CAGD

متصور نبود! برای یک طراح صنعتی یا یک هنرمند طراح که می‌خواست از کامپیوتر استفاده کند، بیش از هر چیز این مهم بود که بتواند مجموعهٔ وسیعی از اشکال و طرحهایی را که رسم می‌کند با یک ابزار واحد مدلسازی کند و سپس با ایجاد تغییرات روی آن بتواند مشخصات هندسی طرح خود را کنترل کند. در این رابطه توابع چندجمله‌ای، ساده‌ترین و تا حدودی کارآمدترین ابزار بودند. با اینحال از دیدگاه هندسی، منحنیها و رویه‌هایی که در پایه‌های معمول چندجمله‌ای نمایش داده می‌شدند، برای کنترل خواص هندسی چندان مناسب نبوند.

اولین تلاشها برای ارائهٔ یک پایه چندجمله‌ای مناسب برای تعابیر هندسی، توسط **دِکاستل‌یائو** [۱۶] [۱۵] و **بزیه** [۲] [۳] و **کُنز** [۷] [۸] در حدود ۱۹۶۰ صورت گرفت، هرچند کارهای ایشان خیلی دیرتر منتشر شد. بدین ترتیب نرم‌افزار $UNISURF$ در زمرهٔ اولین برنامه‌های طراحی هندسی، بر پایهٔ استفاده از منحنی بزیه برای طراحی بدنه و قطعات خودرو، هواپیما، کشتی و ... بکار گرفته شد. دربارهٔ منحنی بزیه در بخش ۱—۴ مطالبی خواهد آمد.

اسپلاینها، توابعی که به طور قطعه‌وار چندجمله‌ای هستند، تعمیم مناسبتر و انتخاب بعدی برای مدلسازی هندسی بودند. توابع پایهٔ Bـاسپلاین بعنوان اسپلاینهایی با ویژگیهای خاص و مناسب برای مدلسازی هندسی، یک پایه برای فضای اسپلاینها تشکیل می‌دهند. این توابع بر خلاف منحنی بزیه، از اواخر قرن نوزدهم در مسائل آنالیز حقیقی و احتمال بکار می‌رفتند. امّا بکارگیری آنها در ۱۹۴۶ توسط **شوئنبرگ** [۳۶]، Bـاسپلاینها را بعنوان یک مبحث مستقل و نوین در نظریهٔ تقریب مطرح کرد. اسپلاینها در دیدگاه اول از دو جهت برای طراحی هندسی مناسب بنظر می‌آمدند. یکی آنکه تقریب زدن توابع روی اسپلاینها تحت شرایط بسیار ضعیفتری نسبت به چندجمله‌ایها، به تابع همگرا می‌شود، و دیگر اینکه رفتار منحنی اسپلاین موضعی و کنترل آن آسانتر است. معرفی توابع Bـاسپلاین به طراحی هندسی، اول بار بوسیلهٔ **ریزنفلد** [۳۴] [۱] و **کاکس** [۹] و **دِبور** [۱۱] صورت گرفت. با توسعهٔ Bـاسپلاینها به Bـاسپلاینهای گویا، مدل $NURBS$[۳] در نرم‌افزارهای طراحی به کمک کامپیوتر بعنوان یک ابزار جامع و فراگیر بکارگرفته می‌شود.

در این فصل به طور مقدماتی و گذرا به معرفی Bـاسپلاینها و نمادگذاریهای بکار رفته در این رساله پرداخته می‌شود و برخی روابط و الگوریتمهای مفید برای Bـاسپلاینها ذکر می‌شود. در پایان منحنی بزیه

[۳] Nonuniform Rational B-Spline

مورد بررسی قرار می‌گیرد.

۱ — ۲ پایه‌های Bـاسپلاین

مطابق معمول در مبحث اسپلاینها، به جای تعیین درجهٔ یک چندجمله‌ای از مرتبهٔ چندجمله‌ای استفاده می‌کنیم. یک چندجمله‌ای را از مرتبهٔ k می‌گوییم اگر درجهٔ آن کمتر از k باشد. فضای توابعی را که روی بازهٔ $[a,b] \subseteq \mathbb{R}$ چندجمله‌ای از مرتبهٔ k هستند، با $\mathcal{P}^k[a,b]$ و در مواردی که ابهامی نداشته باشد با $\mathcal{P}^k$ نشان می‌دهیم. بدین ترتیب با جمع و ضرب اسکالر معمول برای توابع حقیقی، $\mathcal{P}^k$ یک فضای برداری است. برای این فضا پایه‌های متفاوتی می‌توان ارائه کرد، مثل تک جمله‌ایها. امّا در طراحی هندسی عمدتاً توابع پایه برنشتاین مورد توجه هستند که در بخش ۱ — ۴ در ارتباط با منحنی بزیه بکار گرفته می‌شوند.

فضای اسپلاینها شامل توابعی که قطعه‌وار چندجمله‌ای هستند، با ملاحظهٔ برخی نکات به شرح زیر تعریف می‌شود.

تعریف ۱.۱ به ازای

- عدد صحیح و مثبت k،
- عدد صحیح و مثبت n که $n \geq k$،
- دنبالهٔ گره‌های $\mathcal{U} = \{u_i\}_{i=1}^{n+k} \subset \mathbb{R}$ که $u_1 \leq u_2 \leq \cdots \leq u_{n+k}$ و $u_i < u_{i+k}$،

فضای اسپلاینهای مرتبه k روی دنبالهٔ گره‌های $\mathcal{U}$ را با مجموعهٔ $\mathcal{S}_{n,\mathcal{U}}^k$ نشان می‌دهیم و آن شامل توابعی است که روی هر یک از بازه‌های

$$(-\infty\ u_1), [u_1\ u_2), \ldots, [u_{n+k-1}\ u_{n+k}), [u_{n+k}\ \infty)$$

چندجمله‌ای از مرتبه k هستند با این قید که برای هر $f \in \mathcal{S}_{n,\mathcal{U}}^k$ و هر گره u_i داریم

$$f_{left}^{(r)}(u_i) = f_{right}^{(r)}(u_i) \qquad r = 0, 1, \ldots, k-1-\mu_i$$

که در آن $f_{left} = f|_{[u_{i-1}\ u_i)} \in \mathcal{P}^k_{[u_{i-1}\ u_i)}$ وقتی $i > 1$ و $f_{left} = f|_{(-\infty\ u_1)} \in \mathcal{P}^k_{(-\infty\ u_1)}$ وقتی $i = 1$ و

$f_{right} = f|_{[u_i\ u_{i+1})} \in \mathcal{P}^k_{[u_i\ u_{i+1})}$ وقتی $i < n+k$ و $f_{right} = f|_{[u_{n+k}\ \infty)} \in \mathcal{P}^k_{[u_{n+k}\ \infty)}$ وقتی $i = n+k$.

به طور کلی $\mu(t)$ **نشاندهندهٔ حداکثر تعداد تکرار مقدار** t **در دنبالهٔ گره‌هاست** و در اینجا $\mu_i = \mu(u_i)$.

بعبارت دیگر می‌توان گفت که توابع اسپلاین در فاصلهٔ دو گره، چندجمله‌ای مرتبهٔ k و روی هر گره دارای پیوستگی $C^{k-1-\mu_i}$ هستند.

زیرفضای شامل توابع $f \in \mathcal{S}^k_{n,\mathcal{U}}$ که روی **دامنهٔ پارامتر** $\mathcal{I} = [u_k\ u_{n+1})$ تعریف می‌شوند، تشکیل یک فضای برداری با جمع و ضرب اسکالر معمول می‌دهد. وقتی از پایهٔ فضای $\mathcal{S}^k_{n,\mathcal{U}}$ صحبت می‌کنیم منظور همان پایهٔ فضای برداری توابع تحدید شده به بازهٔ $\mathcal{I}$ است.

ساده‌ترین پایه برای این فضا توابع توانی یکطرفه[۴] هستند. یک تابع توانی یکطرفه به صورت

$$(t-x)^r_+ = \begin{cases} 0 & t < x \\ (t-x)^r & t \geq x \end{cases}$$

تعریف می‌شود. پایه مورد بحث، مجموعهٔ تمام توابع توانی یکطرفهٔ زیر است:

$$\begin{array}{llll}
(t-u_{i_1})^{k-1}_+, (t-u_{i_1})^{k-2}_+ & \ldots, & (t-u_{i_1})^{k-\mu_{i_1}}_+ \\
(t-u_{i_2})^{k-1}_+, (t-u_{i_2})^{k-2}_+ & \ldots, & (t-u_{i_2})^{k-\mu_{i_2}}_+ \\
\ldots \quad \ldots & & \\
(t-u_{i_L})^{k-1}_+, (t-u_{i_L})^{k-2}_+ & \ldots, & (t-u_{i_L})^{k-\mu_{i_L}}_+
\end{array}$$

در اینجا L تعداد گره‌های غیر تکراری $\mathcal{U}$، $i_1 = 1$ و برای $j = 2, \ldots, L$، i_j کوچکترین عددی است که $u_{i_j} > u_{i_{j-1}}$. در عمل این پایه برای محاسبات عددی ناپایدار و برای مدلسازی هندسی نامناسب است.

توابع پایه Bـاسپلاین نیز یک پایه برای فضای $\mathcal{S}^k_{n,\mathcal{U}}$ تشکیل می‌دهند. این پایه که پایهٔ Bـاسپلاین $\mathcal{B} = \mathcal{B}^k_{n,\mathcal{U}} = \{B^k_{i,\mathcal{U}}\}^n_{i=1}$ خوانده می‌شود، تنها پایهٔ تشکیل شده از توابعی است، که با دو مشخصهٔ: ۱) دارا بودن کوچکترین تکیه‌گاه توأم با بیشترین مشتقپذیری ممکن در بین سایر توابع اسپلاین و ۲) نرمال شده بقسمی که حاصلجمعشان روی سراسر دامنه برابر واحد باشد، تعیین می‌شوند. چنین توابعی را می‌توان

[۴] One-sided power function

بوسیلهٔ اثر عملگر تفاضلی بر توابع توانی یکطرفه برای رسیدن به یک پایهٔ پایدار بدست آورد:

$$B^k_{i,u}(t) \;=\; (-1)^k (u_{i+k} - u_i)\, [u_i, u_{i+1}, \dots\ , u_{i+k} : x](t-x)^{k-1}_+ \qquad i = 1, 2, \dots, n\ .$$

r اُمین عملگر تفاضلی روی تابع f به صورت

$$[u_i : x] f(x) \;=\; f(u_i)$$

برای $r = 0$ و برای $r \geq 1$ به طور بازگشتی تعریف می‌شود:

$$[u_i, \dots, u_{i+r} : x] f(x) \;=\; \frac{[u_{i+1}, \dots, u_{i+r} : x] f(x) - [u_i, \dots, u_{i+r-1} : x] f(x)}{u_{i+r} - u_i}$$

وقتی $u_{i+r} \neq u_i$ و

$$[u_i, \dots, u_{i+r} : x] f(x) \;=\; \frac{1}{r!}\, D^{(r)}_x\, f(x)|_{x=u_i}$$

وقتی $u_{i+r} = u_i$.

به جای تعریف فوق، یک رابطهٔ اساسی و مستقل از مفهوم توابع توانی یکطرفه را که از حیث محاسبهٔ عددی پایدار است، بعنوان تعریف توابع پایهٔ Bـاسپلاین قرار می‌دهیم.

تعریف ۲.۱ تابع پایهٔ Bـاسپلاین $B^k_{i,\mathcal{U}}(t)$ وابسته به اعداد n, k و دنبالهٔ گره‌های $\mathcal{U}$ که در شرایط تعریف ۱.۱ صدق می‌کنند، برای $i = 1, 2, \dots, n$ به صورت

$$B^1_i(t) \;=\; \begin{cases} 1 & t \in [u_i\ u_{i+1}) \\ 0 & \text{در غیر اینصورت} \end{cases} \qquad (1-1)$$

وقتی $k = 1$ و

$$B^k_i(t) \;=\; \frac{t - u_i}{u_{i+k-1} - u_i}\, B^{k-1}_i(t) \;+\; \frac{u_{i+k} - t}{u_{i+k} - u_{i+1}}\, B^{k-1}_{i+1}(t) \qquad (1-2)$$

وقتی $k > 1$ تعریف می‌شود. در عبارت اخیر اگر $u_{i+k-1} - u_i = 0$ و یا $u_{i+k} - u_{i+1} = 0$، به ترتیب هر یک از ضرایب

$$\frac{t - u_i}{u_{i+k-1} - u_i} \quad , \quad \frac{u_{i+k} - t}{u_{i+k} - u_{i+1}}$$

صفر در نظر گرفته می‌شوند.

هرگاه ابهامی وجود نداشته باشد ممکن است به جای $B^k_{i,\mathcal{U}}(t)$ بنویسیم $B^k_i(t)$ و یا حتیٰ $B_i(t)$.

بدین ترتیب یک تابع اسپلاینی $f \in \mathcal{S}^k_{n,\mathcal{U}}$ را وقتی $f : \mathcal{I} \mapsto \mathbb{R}$، می‌توان به صورت

$$f(t) = \sum_{i=1}^{n} c_i B_i^k(t) \qquad (۳-۱)$$

در پایهٔ Bـاسپلاین نمایش داد. ضرایب c_i در این بسط، ضرایب Bـاسپلاین f گفته می‌شوند. گاهی برای تأکید اینکه چنین نمایشی برای تابع اسپلاینی f در دسترس است، f را یک تابع Bـاسپلاین می‌گوییم.

بر اساس روابطهٔ بازگشتی (۱-۱) و (۱-۲)، که به رابطهٔ **کاکس—دِبور** مشهور است، برخی از ویژگیهای مهم توابع پایهٔ Bـاسپلاین بدست می آیند که بدون آوردن اثبات در اینجا ذکر می‌کنیم (بعنوان یک مرجع کامل و دقیق به [۱۳] نگاه کنید).

ویژگیهای توابع پایهٔ Bـاسپلاین

- توابع پایهٔ Bـاسپلاین نامنفی هستند:

$$B_i^k(t) \geq 0 , \qquad t \in \mathbb{R} . \qquad (۴-۱)$$

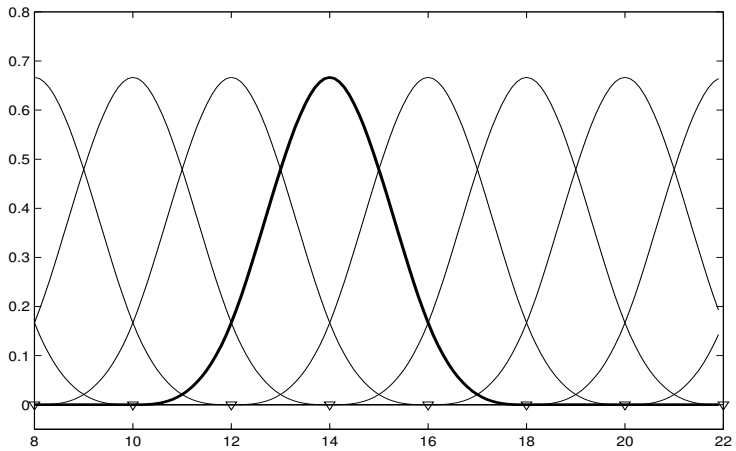
- پایهٔ Bـاسپلاین یک افراز واحد است:

$$\sum_{i=1}^{n} B_i^k(t) = 1 , \quad t \in [u_k \; u_{n+1}) . \qquad (۵-۱)$$

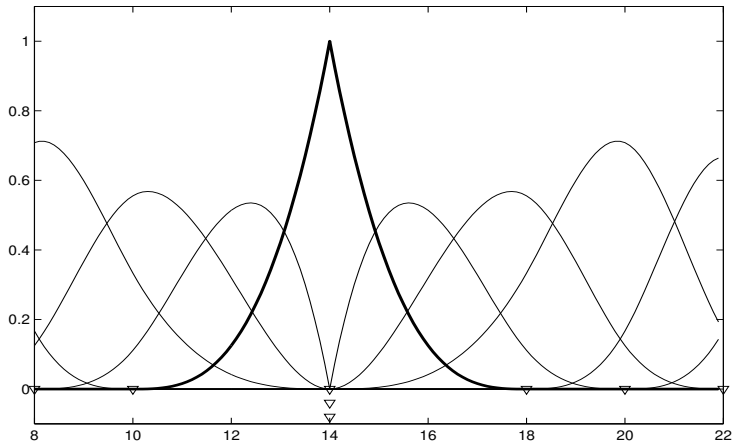
- توابع پایهٔ Bـاسپلاین دارای تکیه‌گاه فشرده هستند:

$$B_i^k(t) = 0 , \quad t \notin [u_i \; u_{i+k}) . \qquad (۶-۱)$$

این ویژگیها توابع Bـاسپلاین را برای کاربرد در طراحی هندسی مناسب می‌سازند. در اشکال ۱-۱ و ۱-۲ نمودار مجموعه توابع پایه Bـاسپلاین بر روی دامنهٔ تعریفشان، روی دو دنباله گره متفاوت نشان داده شده است.



شکل ۱—۱: توابع پایهٔ Bـاسپلاین مرتبه $k = ۴$ روی دنباله گره‌های یکنواخت.



شکل ۱—۲: توابع پایهٔ Bـاسپلاین مرتبه $k = ۴$ روی دنباله‌ای از گره‌های متساوی‌الفاصله با سه بار تکرار گره در مقدار ۱۴.

۱—۳ منحنی Bـاسپلاین

منحنی Bـاسپلاین، یک منحنی فضایی پارامتری است که هر مؤلفهٔ آن یک تابع Bـاسپلاین است.

بعبارت دقیقتر، برای n ضلعی $\{c_i\}_{i=۱}^{n} \subset I\!R^S$، منحنی Bـاسپلاین $Q(t) : \mathcal{I} \mapsto I\!R^S$ به صورت

$$Q(t) = \sum_{i=۱}^{n} c_i B_i^k(t) , \qquad t \in \mathcal{I} = [u_k \; u_{n+۱}) . \tag{۷—۱}$$

تعریف می‌شود. $\{c_i\}_{i=۱}^{n}$ چندضلعی کنترلی منحنی $Q(t)$ خوانده می‌شود.

انتخاب $\mathcal{I}$ بعنوان دامنهٔ پارامتر در اینجا، از این جهت نیز مهم است که خاصیت افراز واحد (۱–۵) تنها روی این دامنه برقرار است. در مباحث طراحی هندسی این ویژگی که ترکیب خطی، یک ترکیب گرانیگاهی باشد مهم است. در یک ترکیب گرانیگاهی $\sum \alpha_i c_i$ داریم $\sum \alpha_i = ۱$. به علاوه اگر برای همهٔ i ها داشته باشیم $\alpha_i \geq ۰$ آنگاه یک ترکیب محدب داریم.

۱–۳–۱ خواص منحنی Bـاسپلاین

قبل از بیان خواص منحنی Bـاسپلاین قرارداد زیر را تا پایان این رساله می‌پذیریم.

قرارداد. برای هر $t \in [u_k \ u_{n+۱})$ عدد δ را چنان تعریف می‌کنیم که

$$u_\delta \leq t < u_{\delta+۱} \ .$$

بدین ترتیب δ بزرگترین اندیسی در دنبالهٔ گره‌ها $\mathcal{U}$ است که مکان درج t را نشان می‌دهد. در مواردی که می‌خواهیم بر وابستگی δ به t تأکید کنیم آنرا به صورت $\delta(t)$ بکار خواهیم برد.

حال ویژگیهای منحنی Bـاسپلاین را بدون ذکر اثبات در زیر می‌آوریم. در اینجا نیز برای اثباتها و مطالب کاملتر به [۱۳] رجوع کنید.

ویژگیهای منحنی Bـاسپلاین

- پایایی تحت تبدیلات آفینی. برای هر تبدیل آفینی[۵] مثل $\Phi : \mathbb{R}^S \mapsto \mathbb{R}^S$ ، داریم

$$\Phi(\sum_{i=۱}^{n} c_i B_i^k(t)) \ = \ \sum_{i=۱}^{n} \Phi(c_i) B_i^k(t) \ . \qquad (۸–۱)$$

- خاصیت پوشش محدب[۶]. تصویر $Q(t)$ در پوشش محدب چندضلعی کنترلی واقع می‌شود:

$$Q(t) \in \mathrm{Cov}(c_۱, c_۲, \ldots, c_n) \ , \quad t \in \mathcal{I} \qquad (۹–۱)$$

که $\mathrm{Cov}(c_۱, \ldots, c_n)$ کوچکترین زیرمجموعه محدب دربردارندهٔ چندضلعی $\{c_i\}_{i=۱}^n$ است.

---

[۵] تبدیل Φ آفینی است اگر $\Phi(\alpha X + \beta Y) = \alpha\Phi(X) + \beta\Phi(Y)$ وقتی $\alpha + \beta = ۱$.

[۶] Convex hull

- خاصیت کنترلپذیری موضعی. هر رأس کنترلی c_i تنها در رفتار یک بخش متناهی از $Q(t)$ مؤثر است. به عبارت دقیقتر، خارج از $[u_i \ u_{i+k}]$ داریم $B_i^k(t) = 0$ و بدین ترتیب تغییرات $Q(t)$ مستقل از c_i است. این خاصیت در شکل ۱—۳ نمایش داده شده است.

- خاصیت پوشش محدب قوی. برای هر $t \in \mathcal{I}$ ، داریم

$$Q(t) = \sum_{i=\delta-k+1}^{\delta-\mu} c_i B_i^k(t) \qquad (۱۰-۱)$$

که $\delta = \delta(t)$ و $\mu = \mu(t)$. بدین ترتیب ملاحظه می‌شود که

$$Q(t) \in \mathrm{Cov}(c_{\delta-k+1}, \ldots, c_{\delta-\mu}) \ . \qquad (۱۱-۱)$$

این خاصیت نیز در شکل ۱—۳ نمایش داده شده است.

- خاصیت کاهش نوسان[۷]. برای دنبالهٔ متناهی $(x_1, \ldots, x_m)$ تابع $V[x_1, \ldots, x_m]$ را برابر تعداد تغییر علامتها در زیر دنبالهٔ حاصل از حذف مؤلفه‌های صفر آن دنباله تعریف می‌کنیم. برای یک تابع حقیقی f تعریف می‌کنیم:

$$V[f] = \sup V[f(x_1), f(x_2), \ldots, f(x_m)]$$

که sup روی تمام دنباله‌های متناهی و صعودی $(x_1, \ldots, x_m)$ و تمام اعداد m گرفته می‌شود.

برای تابع Bـاسپلاین $f \in \mathcal{S}_{n,\mathcal{U}}^k$ با نمایش $f(t) = \sum c_i B_i^k(t)$ داریم

$$V[f] \leq V[c_1, c_2, \ldots, c_n] \ . \qquad (۱۲-۱)$$

در مورد منحنی Bـاسپلاین $Q(t)$ گزارهٔ فوق به این شکل بیان می‌شود:

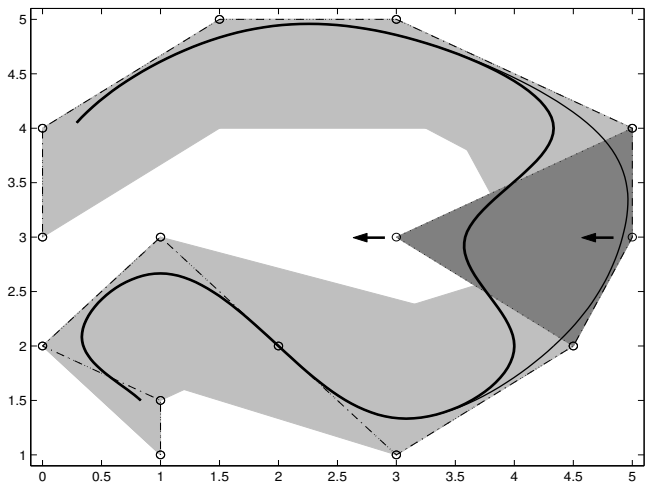
فرض کنید $k > 1$ و $V[\ell\,; c_1, \ldots, c_n]$ و $V[\ell\,; Q]$ به ترتیب تعداد نقاط تقاطع خط $\ell \subset \mathbb{R}^S$ با چندضلعی $\{c_i\}_{i=1}^n$ و منحنی Q باشد. آنگاه برای منحنی Bـاسپلاین $Q(t)$ باچندضلعی کنترلی $\{c\}_{i=1}^n$ و هر خط دلخواه ℓ ، داریم [۱۴]

$$V[\ell\,; Q] \leq V[\ell\,; c_1, c_2, \ldots, c_n] \ . \qquad (۱۳-۱)$$

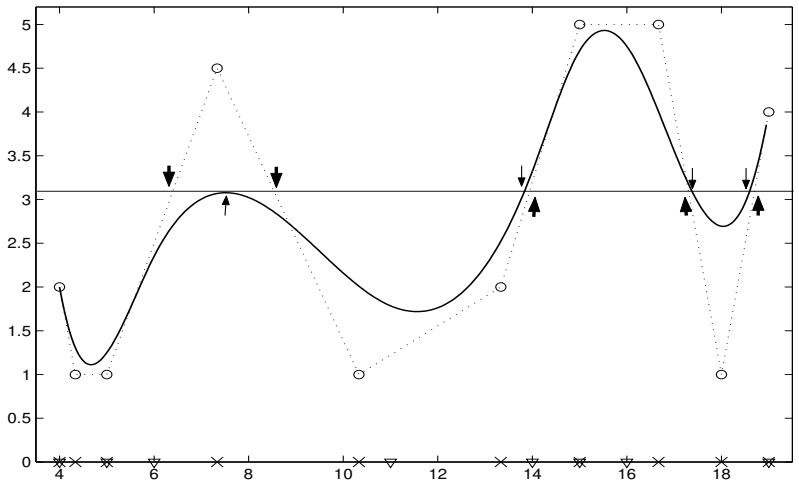
در شکل ۱—۴ نمونه‌ای از این ویژگی نشان داده می‌شود.

[۷] Variation diminishing property

- مشتقپذیری. برای یک دنبالهٔ ساده از گره‌ها (بدون گره تکراری)، منحنی $Q(t)$ دارای پیوستگی C^{k-2} در سراسر دامنهٔ تعریف خود است. با داشتن گره‌هایی با مرتبه تکرار بیشتر از یک، به همان اندازه از مشتقپذیری $Q(t)$ روی آن گره کاسته می‌شود.



شکل ۱—۳: کنترلپذیری موضعی و پوشش محدب قوی در یک منحنی Bـاسپلاین.



شکل ۱—۴: کاهش نوسان در Bـاسپلاینها: یک خط راست نمودار تابع Bـاسپلاین را بیش از چندضلعی دِبور آن قطع نمی‌کند.

وجود این ویژگیها برای منحنی Bـاسپلاین مهم است. به طور مثال، دارا بودن ویژگی پایایی تحت تبدیلات آفینی مثل دوران، انتقال و غیره، برای انتخاب یک مدل در CAD یک اصل ضروری است. سایر ویژگیها نیز در عمل مفید واقع می‌شوند. مثلاً بر پایهٔ خاصیت کنترلپذیری موضعی، طراح اطمینان خواهد داشت که تغییر یک رأس کنترلی تنها یک تغییر موضعی روی منحنی ایجاد می‌کند و سایر قسمتهای منحنی

ثابت باقی می‌ماند، بر پایهٔ خاصیت پوشش محدب قوی می‌تواند پیش‌بینی کند که منحنی از چه جاهایی می‌گذرد و بر اساس خاصیت کاهش نوسان و مشتقپذیری منحنی می‌تواند انتظار داشته باشد که منحنی، تقریبی هموارشده از چندضلعی کنترلی است.

۱–۳–۲ ترکیب توابع پایهٔ Bـاسپلاین

با بکارگیری روابط بازگشتی (۱–۱) و (۱–۲) روی ترکیب $\sum c_i B_i^k$ و انجام کمی محاسبات می‌توان یک رابطهٔ بازگشتی نیز برای محاسبهٔ منحنی Bـاسپلاین به صورت زیر بدست آورد:

$$Q(t) = \sum_{i=1}^{n} c_i B_i^k(t) = \sum_{i=1}^{n+1} \left(\frac{t-u_i}{u_{i+k-1}-u_i}\, c_i + \big(1 - \frac{t-u_i}{u_{i+k-1}-u_i}\big)\, c_{i-1} \right) B_i^{k-1}(t) \qquad (۱۴–۱)$$

که در اینجا $c_0 = c_{n+1} = 0$ فرض می‌شود. اگر قرار دهید

$$w_i^{[r]} := \frac{t-u_i}{u_{i+k-r}-u_i} \qquad (۱۵–۱)$$

و

$$c_i^{[1]} := c_i \ , \qquad i = 1, \ldots, n$$
$$c_0^{[r]} = c_{n+r}^{[r]} := 0 \ , \qquad r = 1, 2, \ldots, k-1$$

$$c_i^{[r+1]} := (1 - w_i^{[r]})\, c_{i-1}^{[r]} + w_i^{[r]}\, c_i^{[r]} \ , \qquad \begin{cases} r = 1, \ldots, k-1 \\ i = 1, \ldots, n+r-1 \end{cases} \qquad (۱۶–۱)$$

آنگاه برای هر $r = 1, \ldots, k$ ، داریم

$$Q(t) = \sum_{i=1}^{n+r-1} c_i^{[r]}\, B_i^{k+1-r}(t) \ . \qquad (۱۷–۱)$$

به خصوص به ازای $r = k$ و با توجه به اینکه برای $t \in [u_\delta \ u_{\delta+1})$، تمام توابع پایهٔ $B_i^1(t) = 0$ بجز $B_\delta^1(t) = 1$، نتیجه می‌گیریم $Q(t) = c_\delta^{[k]}$. این یک روش مناسب برای پارامتری کردن منحنی Bـاسپلاین است.

۱–۳–۳ مشتق

مشتق توابع پایهٔ Bـاسپلاین و منحنی Bـاسپلاین به صورت یک رابطهٔ جالب بدست می‌آیند[۱۳]:

$$D\,B_i^k(t) \;=\; (k-1)\left(\frac{B_i^{k-1}(t)}{u_{i+k-1}-u_i} - \frac{B_{i+1}^{k-1}(t)}{u_{i+k}-u_{i+1}}\right)\,. \qquad (۱۸–۱)$$

در اینجا نیز هر کسر با مخرج صفر را برابر صفر در نظر می‌گیریم. بدین ترتیب مشاهده می‌شود که مشتق یک تابع Bـاسپلاین مرتبهٔ k به صورت ترکیب تفاضلی دو تابع Bـاسپلاین مرتبهٔ $k-1$ بیان می‌شود. این ویژگی به طور مشابه به منحنی Bـاسپلاین نیز منتقل می‌شود:

$$D\left(\sum_{i=1}^{n} c_i B_i^k(t)\right) \;=\; \sum_{i=1}^{n+1} d_i B_i^{k-1}(t) \qquad (۱۹–۱)$$

که ضرایب منحنی Bـاسپلاین مشتق به صورت

$$d_i \;=\; (k-1)\left(\frac{c_i - c_{i-1}}{u_{i+k-1}-u_i}\right)\,, \quad i = 1, 2, \ldots, (n+1) \qquad (۲۰–۱)$$

تعریف می‌شوند. در اینجا نیز $c_0 = c_{n+1} = 0$ فرض می‌شوند و وقتی $u_{i+k-1} = u_i$ مقدار d_i برابر صفر قرار داده می‌شود.

یک تعبیر هندسی جالب و بسیار مهم که در ارتباط با همین موضوع بدست می‌آید بدین قرار است که هر تابع Bـاسپلاین را وقتی $k > 1$، می‌توان به صورت یک منحنی Bـاسپلاین بیان کرد:

$$\begin{pmatrix} t \\ f(t) = \sum_{i=1}^{n} c_i B_i^k(t) \end{pmatrix} \;=\; \sum_{i=1}^{n} \begin{pmatrix} \xi_i \\ c_i \end{pmatrix} B_i^k(t)\,, \qquad t \in \mathcal{I}\,. \qquad (۲۱–۱)$$

توجه کنید که طرف چپ، نمودار یک تابع حقیقی و طرف راست یک منحنی در $\mathbb{R}^2$ است. نقاط $\begin{pmatrix} \xi_i \\ c_i \end{pmatrix}$ یک چندضلعی تشکیل می‌دهند که به چندضلعی دِبور مشهور است و نقاط ξ_i موسوم به نقاط گرویل[۸] به عنوان میانگین گره‌ها به صورت زیر تعریف می‌شوند:

$$\xi_i \;=\; \frac{1}{k-1}\sum_{j=i+1}^{i+k-1} u_j\,, \qquad i = 0, 1, 2, \ldots, n, n+1\,. \qquad (۲۲–۱)$$

ξ_i ها در اصل ضرایب بسط Bـاسپلاین تابع همانی t در پایهٔ $\mathcal{B}_{n,\mathcal{U}}^k$ هستند و رابطهٔ $t = \sum_{i=1}^{n} \xi_i B_i^k(t)$، ویژگی دقت خطی پایهٔ Bـاسپلاین را نشان می‌دهد. در شکل ۱–۴ چندضلعی دِبور یک تابع Bـاسپلاین

[۸] Greville abscissae

همانند چندضلعی کنترلی منحنی نمودار آن تابع نشان داده شده است. در این شکل نقاط گرویل با علامت $\times$ مشخص شده‌اند، در حالیکه $\circ$ ها گره‌ها را نشان می‌دهند.

حال می‌توان نشان داد که

$$d_i = \frac{c_i - c_{i-1}}{\xi_i - \xi_{i-1}}, \qquad i = 1, 2, \ldots, n+1. \tag{۲۳–۱}$$

تعبیر هندسی این رابطه اینست که ضرایب Bـاسپلاین مشتق d_i، برابر با شیب اضلاع چندضلعی دِبور است. نقاط گرویل بعد از این نیز به کار خواهند رفت.

۱–۳–۴ تولید منحنی‌های آزاد

برای استفاده از منحنیهای Bـاسپلاین در تولید منحنی‌های آزاد[۹] هر سه عامل چندضلعی کنترلی $\{c_i\}_{i=1}^{n}$، دنبالهٔ گره‌ها $\mathcal{U}$ و مرتبهٔ Bـاسپلاین k به کار می‌آیند. چندضلعی کنترلی عمدتاً بوسیلهٔ طراح داده می‌شود ولی در حالت کلی باید چنان باشد که با توجه به خواص منحنی Bـاسپلاین، محدودهٔ تقریبی تغییرات آن را کنترل کند. چند حالت خاص در وضعیت رئوس کنترلی قابل توجه هستند.

۱) درونیابی با تکرار رأس کنترلی. با $k-1$ بار تکرار یک رأس کنترلی، منحنی Bـاسپلاین از آن رأس می‌گذرد. با تکرار رأس از پیوستگی منحنی کاسته نمی‌شود، امّا برخی مشتقات منحنی در آن رأس صفر می‌شود و منحنی گوشه‌دار می‌شود. به عبارت دقیقتر، اگر $c_i = c_{i+1} = \cdots = c_{i+k-2}$، منحنی ضمن گذر از c_i، به ترتیب با دو ضلع $c_{i-1}c_i$ و $c_{i+k-2}c_{i+k-1}$ مماس می‌شود.

۲) تقریب خطی با رئوس کنترلی همخط. اگر $2k-2$ رأس متوالی $c_{i-k+1}, \ldots, c_{i-1}, c_i, \ldots, c_{i+k-2}$ همخط باشند، منحنی تمام ضلع $c_{i-1}c_i$ را طی می‌کند.

۳) منحنی بسته با رویهم‌گذاری رئوس ابتدا و انتها. اگر دنبالهٔ گره‌ها یکنواخت باشد با رویهم قرار دادن $k-1$ رأس اول و آخر یک منحنی تناوبی بدست می‌آید.

[۹] Free form curve

مرتبهٔ Bـاسپلاین مورد استفاده نوعاً به شرایط طراحی بستگی دارد. حالتهای $k = ۳, ۴$ (Bـاسپلاین مربعی و مکعبی) بیشتر مورد استفاده قرار می‌گیرند. حالت $k = ۲$ (Bـاسپلاین خطی) به طور بدیهی فقط چندضلعی کنترلی را درونیابی می‌کنند. حالت $k = ۳$ با آنکه حداکثر یک مرتبه مشتقپذیر است، امّا در $\mathbb{R}^۳$ هر یک از قطعات منحنی Bـاسپلاین مربعی، در یک صفحهٔ مسطح و معمولاً متقاطع با یکدیگر قرار دارند و این امر در برخی کاربردها مطلوب نیست. با افزایش k به وضوح میزان مشتقپذیری منحنی بالاتر می‌رود ولی اثر موضعی رئوس کنترلی پخش می‌شود و خاصیت پوشش محدب قوی باعث می‌شود منحنی با آزادی بیشتری درون چندضلعی کنترلی رفتار کند. اساساً در مورد اسپلاینها برای حصول تقریب بهتر، به جای افزایش مرتبهٔ اسپلاین k، تعداد داده‌های نمونه n را افزایش می‌دهیم.

اثر گره‌ها

انتخاب گره‌ها برای یک منحنی Bـاسپلاین در برخی موارد دارای اهمیت است. در کاربردهایی که کاربر به عنوان یک طراح، تنها با رئوس کنترلی سروکار دارد می‌توان با انتخاب اتوماتیک گره‌ها او را از این کار معاف کرد. در این موارد معمولاً دنبالهٔ گره‌ها به طور یکنواخت $u_i = i$ انتخاب می‌شوند. شکل منحنی به اندازهٔ $\Delta u_i = u_i - u_{i-۱}$ و مقدار $u_۱$ بستگی ندارد، بلکه به نسبت $\frac{\Delta u_i}{\Delta u_{i+۱}}$ بستگی دارد. به طور کلی می‌دانیم که تکرار یک گره باعث کاهش مشتقپذیری منحنی در آن گره می‌شود. وقتی $\mu_i = k - ۱$، منحنی در u_i تنها پیوسته $C^۰$ خواهد بود و می‌توان نشان داد هر منحنی Bـاسپلاین روی این دنبالهٔ گره‌ها رأس $c_{\delta(u_i)-k+۱}$ را درونیابی می‌کند. اگر $\mu_i = k$، منحنی در u_i ناپیوسته می‌شود امّا $Q(u_i^-) = c_{\delta(u_i)-k}$ و $Q(u_i^+) = c_{\delta(u_i)-k+۱}$.

گاهی از این ویژگی برای درونیابی برخی رئوس کنترلی استفاده می‌شود، به خصوص یک حالت بسیار متداول، انتخاب k گره اول و آخر برابر با u_k و $u_{n+۱}$ است:

$$u_۱ = \cdots = u_k \quad , \quad u_{n+۱} = \cdots = u_{n+k} \; . \qquad (۲۴-۱)$$

این حالت برای دنبالهٔ گره‌ها را بعداً نیز مورد استفاده قرار می‌دهیم. در این حالت همیشه رئوس اول و آخر درونیابی می‌شوند.

یک انتخاب متناسب با طول وتر نیز برای گره‌ها در کاربردهای هندسی مورد استفاده است. این انتخاب باید به گونه‌ای باشد که

$$\frac{\xi_i - \xi_{i-1}}{\xi_{i+1} - \xi_i} = \frac{\|c_i - c_{i-1}\|_2}{\|c_{i+1} - c_i\|_2} .$$

مثلاً با ثابت نگهداشتن $u_2, \ldots, u_k$ و انتخاب دلخواه برای u_1, u_{n+k} و قرار دادن

$$u_{i+k-1} = u_i + \|c_i - c_{i-1}\|_2 , \qquad i = 2, \ldots, n$$

می‌توان یک دنبالهٔ گره متناسب با طول وتر داشت.

۱—۴ منحنی بزیه

توابع پایهٔ برنشتاین که به صورت

$$P_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i} , \qquad i = 0, 1, \ldots, n$$

تعریف می‌شوند یک پایه برای $\mathcal{P}^{n+1}[0\ 1]$ می‌سازند. به وضوح مشاهده می‌شود که این توابع روی دامنهٔ تعریف $[0\ 1]$ اولاً نامنفی هستند و ثانیاً دارای جمع واحد هستند:

$$\sum_{i=0}^{n} P_i^n(t) = \sum_{i=0}^{n} \binom{n}{i} t^i (1-t)^{n-i} = (t + (1-t))^n = 1 .$$

منحنی بزیه $Q : [0\ 1] \mapsto \mathbb{R}^S$ از درجهٔ n وابسته به $n+1$ نقطهٔ $\{c_i\}_{i=0}^n \subset \mathbb{R}^S$ به صورت

$$Q(t) = \sum_{i=0}^{n} c_i P_i^n(t) , \qquad t \in [0\ 1]$$

تعریف می‌شود. $\{c_i\}_{i=0}^n$ چندضلعی کنترلی منحنی بزیهٔ $Q(t)$ نامیده می‌شود.

۱—۴—۱ خواص منحنی بزیه

برخی خواص منحنی بزیه عبارتند از[۲۱]:

- خاصیت پوشش محدب.

$$Q(t) \in \mathrm{Cov}(c_0, c_1, \ldots, c_n) \ , \qquad t \in [0\ 1] \ .$$

- خاصیت مجازی کنترلپذیری موضعی[10]. هر تابع $P_i^n(t)$ روی $[0\ 1]$ تنها در نقطهٔ $t = \frac{i}{n}$ به ماکسیمم خود می‌رسد. این سبب می‌شود که رأس کنترلی c_i حداکثر اثر خود را بر رفتار منحنی در اطراف این نقطه داشته باشد[21].

- خاصیت کاهش نوسان. برای خط دلخواه $\ell \subset I\!R^S$ داریم

$$V[\ell\,; Q] \leq V[\ell\,; c_0, c_1, \ldots, c_n] \ .$$

- درونیابی نقاط و شیب در ابتدا و انتها.

$$P_i^n(0) = 1 \iff i = 0 \quad \Longrightarrow \quad Q(0) = c_0 \ ,$$

$$P_i^n(1) = 1 \iff i = n \quad \Longrightarrow \quad Q(1) = c_n \ .$$

مشتق منحنی بزیه به صورت زیر محاسبه می‌شود:

$$D\,P_i^n(t) \ = \ n\,(P_{i-1}^{n-1}(t) - P_i^{n-1}(t))$$

$$D\,Q(t) \ = \ n \sum_{i=0}^{n-1} (c_{i+1} - c_i) P_i^{n-1}(t) \ .$$

بدین ترتیب مشاهده می‌شود که منحنی بزیه به طور مماس با ضلع $c_0 c_1$ شروع می‌شود و به طور مماس با ضلع $c_{n-1} c_n$ خاتمه می‌یابد:

$$D\,Q(0) = n(c_1 - c_0) \quad , \quad D\,Q(1) = n(c_n - c_{n-1}) \ .$$

یکی از ویژگیهای جالب منحنی‌های بزیه و Bـاسپلاین قابلیت تعبیر یکی بر حسب دیگری است. دو قضیهٔ زیر این ارتباط دوطرفه را که در تبادل مدلهای هندسی بین برنامه‌های متفاوت کاربرد زیادی دارد، تشریح می‌کنند.

---

[10] Pseudo-local control

قضیه‌ی ۳.۱ منحنی بزیه به عنوان یک منحنی Bـاسپلاین. منحنی بزیهٔ درجه n با چندضلعی کنترلی $\{c_i\}_{i=0}^{n}$، با منحنی Bـاسپلاین مرتبهٔ $k = n + 1$ تعریف شده روی دنبالهٔ گره‌های $u_1 = \cdots = u_k = 0$, $u_{k+1} = \cdots = u_{2k} = 1$ و با همان چندضلعی کنترلی، برابر است.

قضیه‌ی ۴.۱ منحنی Bـاسپلاین به عنوان چند منحنی بزیه پیوسته. تصویر منحنی Bـاسپلاین مرتبهٔ $k > 1$ با رئوس کنترلی $c_1, \ldots, c_n$ که $n = 1 + L(k-1)$، روی دنبالهٔ گره‌هایی که مرتبهٔ تکرار گره‌های اول و آخر آن k و مرتبهٔ تکرار مابقی گره‌های آن $k-1$ است، با تصویر دنباله‌ای از L منحنی بزیه درجه $k-1$ با چندضلعی‌های کنترلی $c_1, c_2, \ldots, c_k$ و $c_k, c_{k+1}, \ldots, c_{2k-1}$ و $\ldots$ و $c_{n-k+1}, c_{n-k+2}, \ldots, c_n$، برابر است.

برهان. برای یک اثبات قضیهٔ اول به [۱۳] یا [۳۳] و برای اثبات قضیهٔ دوم به [۲۱] رجوع کنید. □

همانطور که از خواص منحنی بزیه دیده می‌شود و طبق آنچه در قضیهٔ تقریب وایراشتراوس در مباحث کلاسیک اثبات می‌شود، در مورد منحنی بزیه نیز همانند منحنی Bـاسپلاین می‌توان گفت که این منحنی تقریب هموارشده‌ای از چندضلعی کنترلی خود است. این فصل را با تشریح الگوریتمهایی برای تظریف منحنی بزیه به پایان می‌رسانیم.

۱—۴—۲ تظریف منحنی بزیه

هدف از تظریف، بدست آوردن یک چندضلعی کنترلی جدید از چندضلعی کنترلی اولیه با تعداد رئوس کنترلی بیشتر، برای رسیدن به یک چندضلعی نزدیکتر به منحنی است. این عمل در منحنیهای بزیه به دو صورت قابل تصور است. در روش اول یک منحنی بزیه به دو منحنی بزیه کوچکتر از همان درجه تجزیه می‌شود. در روش دوم با بالا بردن درجهٔ منحنی بزیه، تعداد رئوس کنترلی یک واحد اضافه می‌شود و چندضلعی به منحنی نزدیکتر می‌شود.

الگوریتم دِکاستل‌یائو

۱) منحنی بزیهٔ Q با رئوس کنترلی $c_0, c_1, \ldots, c_n$ داده شده است. یک $t \in [0\ 1]$ دلخواه و ثابت نگهداشته شده را در نظر بگیر.

۲) قرار بده $r = 0$ و برای $i = 0, 1, \ldots, n$ قرار بده $c_i^{[0]} = c_i$.

۳) برای $r = 1, \ldots, n$ تکرار کن:

$$c_i^{[r]} = (1-t)\, c_i^{[r-1]} + t\, c_{i+1}^{[r-1]}, \qquad i = 0, 1, \ldots, n-r\,.$$

۴) حال تصویر منحنی Q برابر است با اجتماع تصویر دو منحنی بزیهٔ Q^1 با چندضلعی کنترلی $c_0^{[0]}, c_0^{[1]}, \ldots, c_0^{[n]}$ و Q^2 با چندضلعی کنترلی $c_0^{[n]}, c_1^{[n-1]}, \ldots, c_n^{[0]}$:

$$Q_{[c_0, c_1, \ldots, c_n]}([0\ 1]) = Q^1_{[c_0^{[0]}, c_0^{[1]}, \ldots, c_0^{[n]}]}([0\ 1]) \cup Q^2_{[c_0^{[n]}, c_1^{[n-1]}, \ldots, c_n^{[0]}]}([0\ 1])\,.$$

توجه کنید که $c_n^{[0]} = c_n, c_0^{[0]} = c_0$ و به علاوه می‌توان نشان داد که $Q(t) = Q^1(1) = Q^2(0) = c_0^{[n]}$.

در بعضی از مراجع این الگوریتم را به عنوان تعریف منحنی بزیه در نظر می‌گیرند [۲۱]. معمولاً با انتخاب $t = 1/2$ از این الگوریتم برای تظریف منحنی بزیه استفاده می‌شود. با تغییر t روی $[0\ 1]$ و اجرای مکرر الگوریتم فوق و تنها استفاده از اینکه $Q(t) = c_0^{[n]}$، می‌توان منحنی بزیه را به طور عددی و پایدار، پارامتری نمود. این الگوریتم را می‌توان به الگوریتمی برای تجزیه منحنی بزیه به سه زیر منحنی نیز تعمیم داد[۲۱].

بالا بردن درجه

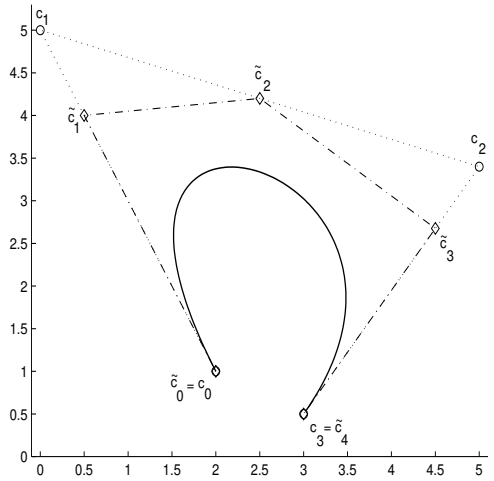
از آنجا که $\mathcal{P}^{n+1}[0\ 1]$ یک زیرفضای برداری از $\mathcal{P}^{n+2}[0\ 1]$ است، یک منحنی بزیهٔ نمایش یافته در آن را می‌توان به صورت یک منحنی بزیه در فضای از مرتبهٔ بالاتر نیز نمایش داد. برای منحنی بزیهٔ Q از درجهٔ n با رئوس کنترلی $c_0, c_1, \ldots, c_n$، ضرایب تبدیل پایه از فضای درجهٔ n به فضای درجهٔ $n+1$ به صورت زیر

روی رئوس کنترلی عمل می‌کنند:

$$\tilde{c}_i = \frac{i}{n+1} c_{i-1} + \left(1 - \frac{i}{n+1}\right) c_i \,, \qquad i = 0, 1, \ldots, (n+1) \qquad (1-25)$$

که $\{\tilde{c}_i\}_{i=0}^{n+1}$ چندضلعی کنترلی Q در پایهٔ جدید است.

شکل ۱—۵ بالا بردن درجه در یک منحنی بزیه درجه ۳ و تبدیل رئوس کنترلی آن به رئوس کنترلی یک منحنی بزیه درجه ۴ را نشان می‌دهد.



شکل ۱—۵: افزایش درجه در منحنی بزیه.

کاربرد الگوریتم فوق بیشتر در تبادل داده‌های هندسی مدل شده، بین برنامه‌هایی است که هر یک تنها قادر به کار با منحنیهای بزیه با یک درجهٔ ثابت هستند. این تبدیل از یک فضای درجه پایین به فضای از درجه بالاتر با یک یا چند بار تکرار الگوریتم فوق انجام پذیر است. امّا برعکس، وقتی می‌خواهیم از فضای درجه بالا به فضای از درجهٔ پائینتر برویم، همیشه این مشکل وجود دارد که این تبدیل به طور یکتا، معکوس‌پذیر نیست. برای معکوس‌سازی این تبدیل که هدف اصلی در این رساله بررسی آن البته در مورد Bـاسپلاینهاست، در بخش ۳—۱—۱ مطالب مختصری آورده خواهد شد.

فصل ۲

درج گره

۲–۱ کلیات

ایدهٔ درج گره در Bـاسپلاینها از آنجا مورد توجه واقع می‌شود که این کار اساساً منجر به افزایش تعداد داده‌های نمونه یا به عبارت دیگر تظریف چندضلعی کنترلی منحنی می‌شود و بدین ترتیب چندضلعی کنترلی یک تقریب بهتر از منحنی را نشان می‌دهد. یک دیدگاه شهودی چنین است که وقتی بُعد فضایی که تابع را روی آن تقریب می‌زنیم بالاتر رود، خطای تقریب کاهش می‌یابد. در مورد $\mathcal{P}^k[a\ b]$ این کار با استفاده از چندجمله‌ایهای از درجه یا مرتبهٔ بالاتر صورت می‌گیرد: $\dim(\mathcal{P}^k) = k$، در حالیکه $\dim(\mathcal{S}^k_{n,\mathcal{U}}) = n$ و افزودن گره به $\mathcal{U}$ سبب افزایش بُعد فضا n می‌شود. در مبحث درج گره علاقمندیم روابطی برای تبدیل توابع نمایش یافته در $\mathcal{S}^k_{n,\mathcal{U}}$ به $\mathcal{S}^k_{m,\tilde{\mathcal{U}}}$ جستجو کنیم در حالیکه $\tilde{\mathcal{U}}$ دنبالهٔ گره‌های حاصل از درج یک یا چند گره در $\mathcal{U}$ است. وقتی $\mathcal{U} \subset \tilde{\mathcal{U}}$ آنگاه $\mathcal{S}^k_{n,\mathcal{U}} \subset \mathcal{S}^k_{m,\tilde{\mathcal{U}}}$. بدین ترتیب روابط مورد بحث با بدست آوردن ضرایب ماتریس تبدیل پایه مشخص می‌شود.

برای قرارداد فرض کنید $\tilde{\mathcal{U}}$ دنبالهٔ گره‌های جدید، بعد از درج برخی گره‌ها در $\mathcal{U}$ باشد و تابع $\tilde{B}^k_i(t) \in \mathcal{B}^k_{m,\tilde{\mathcal{U}}}$ یک عنصر پایهٔ Bـاسپلاین فضای جدید باشد. برای اختصار قرار می‌دهیم $\mathcal{B} = \mathcal{B}^k_{n,\mathcal{U}}$ و $\tilde{\mathcal{B}} = \mathcal{B}^k_{m,\tilde{\mathcal{U}}}$. بدین ترتیب $B^k_i(t) \in \mathcal{B}$ و $\tilde{B}^k_i(t) \in \tilde{\mathcal{B}}$. اگر قرار دهیم

$$n = \dim(\mathcal{S}^k_{n,\mathcal{U}}) \quad , \qquad m = \dim(\mathcal{S}^k_{m,\tilde{\mathcal{U}}}) \tag{۱–۲}$$

$$\mathbf{B}_L = \begin{pmatrix} B_1^k \\ \vdots \\ B_n^k \end{pmatrix} \quad , \qquad \tilde{\mathbf{B}}_L = \begin{pmatrix} \tilde{B}_1^k \\ \vdots \\ \tilde{B}_m^k \end{pmatrix} \tag{۲–۲}$$

آنگاه یک ماتریس رتبه کامل A_L^T با ابعاد $n \times m$ موجود است که

$$A_L^T \tilde{\mathbf{B}}_L \;=\; \mathbf{B}_L \;. \tag{۳–۲}$$

اگر $f \in \mathcal{S}_{n,\mathcal{U}}^k$ دارای نمایش $f = \sum_{i=1}^{n} c_i B_i^k$ باشد، آنگاه بوضوح دارای نمایش $f = \sum_{i=1}^{m} \tilde{c}_i \tilde{B}_i^k$ در $\mathcal{S}_{m,\tilde{\mathcal{U}}}^k$ است که ضرایب c_i و $\tilde{c}_i$ با رابطهٔ

$$\mathbf{c}_L = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} \quad , \quad \tilde{\mathbf{c}}_L = \begin{pmatrix} \tilde{c}_1 \\ \vdots \\ \tilde{c}_m \end{pmatrix}$$

$$f = \mathbf{B}_L^T \mathbf{c}_L \quad , \quad f = \tilde{\mathbf{B}}_L^T \tilde{\mathbf{c}}_L \tag{۴–۲}$$

$$\tilde{\mathbf{c}}_L \;=\; A_L \mathbf{c}_L \tag{۵–۲}$$

به هم مربوط می‌شوند. بدین ترتیب ملاحظه می‌شود که با یافتن ضرایب ماتریس A_L و انجام تبدیل فوق می‌توان چندضلعی کنترلی یک منحنی Bـاسپلاین را تا میزان مورد نظر تظریف کرد. درباره عکس این عمل، یعنی رسیدن از $\tilde{\mathbf{c}}_L$ به $\mathbf{c}_L$ نیز در فصل آینده بحث می‌کنیم.

۲–۲ درج تک گره

در این بخش یک حالت مقدماتی را که درج تک گره است، بررسی می‌کنیم و از آن به عنوان مبنای درج چند گره در حالت کلی بهره‌برداری خواهیم کرد. در اینجا به طور مشخص هدف درج یک گره جدید با مقدار t (منظور پارامتر نیست!)، در $\mathcal{U}$ است به طوری که

$$u_k \le u_\delta \le t < u_{\delta+1} < u_{n+1} \tag{۶–۲}$$

و $\mu = \mu(t) < k$ مرتبهٔ تکرار t در $\mathcal{U}$ قبل از درج است:

$$u_{\delta-\mu+1} = \cdots = u_\delta = t\,. \qquad (۷-۲)$$

دو قید $k \le \delta < n+1$ و $\mu < k$ را تا پایان این رساله در نظر می‌گیریم. اولی تأکید می‌کند که گره t یک گره داخلی باشد (منظور داخل دامنهٔ تعریف $\mathcal{I}$) و دومی تأکید می‌کند که گره t در $\mathcal{U}$ اشباع نشده باشد. در بسیاری از موارد با انتخاب دنبالهٔ گره‌هایی با گره‌های اول و آخر k بار تکرار شده و با رعایت قید $\mu < k$، قید درونی بودن گره خودبخود ارضاء می‌شود. بدین ترتیب به طور کلی داریم

$$\begin{aligned} \tilde{\mathcal{U}} &= \mathcal{U} \cup \{t\} \\ m &= n+1 \\ \tilde{\delta} = \delta + 1 \quad &; \quad \tilde{\mu} = \mu + 1 \end{aligned}$$

$$\tilde{u}_i = \begin{cases} u_i & 1 \le i \le \delta \\ t & i = \delta + 1 \\ u_{i-1} & \delta + 2 \le i \le n + k + 1 \end{cases} \qquad (۸-۲)$$

برای محاسبهٔ ضرایب تبدیل به دو لم زیر نیاز داریم.

لم ۱.۲ برای $k > 1$ و هر تابع Bـاسپلاین $B_i^k(t)$ روی گره‌های $u_1, \ldots, u_{n+k}$ داریم

$$B_i^k(t) = \frac{(t-u_i)^{k-1}}{(u_{i+k-1}-u_i)\cdots(u_{i+1}-u_i)}\,, \quad t \in [u_i\ u_{i+1}) \text{ وقتی} \qquad (۹-۲)$$

و

$$B_i^k(t) = \frac{(u_{i+k}-t)^{k-1}}{(u_{i+k}-u_{i+1})\cdots(u_{i+k}-u_{i+k-1})}\,, \quad t \in [u_{i+k-1}\ u_{i+k}) \text{ وقتی} \qquad (۱۰-۲)$$

در هر دو رابطه اگر هریک از عوامل مخرج صفر شوند، مقدار کسر صفر فرض می‌شود.

برهان. با استفاده از روابط بازگشتی (۱–۱) و (۱–۲) و توجه به اینکه برای هر $t \in [u_i\ u_{i+1})$، توابع Bـاسپلاین $B_{i+1}^{k-1}(t), B_{i+1}^{k-2}(t), \ldots, B_{i+1}^1(t)$، صفر هستند و برای هر $t \in [u_{i+k-1}\ u_{i+k})$، توابع

Bـاسپلاین $B_i^{k-1}(t), B_{i+1}^{k-2}(t), \ldots, B_{i+k-2}^{1}(t)$، صفر هستند، لم اثبات می‌شود. □

حال فرض کنید گره جدید t را در $\mathcal{U}$ درج کردیم و $\tilde{\mathcal{U}}$ بدست آمد. طبق رابطهٔ (۲–۳) داریم

$$B_i^k = \sum_{j=1}^{n+1} \alpha_{i,j}\, \tilde{B}_j^k \ , \qquad i = 1, \ldots, n \tag{۲–۱۱}$$

که $\alpha_{i,j}$ ضرایب ماتریس A_L^T با ابعاد $n \times (n+1)$ هستند. انتظار می‌رود که بسیاری از این ضرایب صفر باشد.

لم ۲.۲ با توجه به مفروضات فوق داریم

$$\tilde{B}_i^k = B_i^k \ , \qquad i = 1, \ldots, \delta - k \ , \tag{۲–۱۲}$$

$$\tilde{B}_i^k = B_{i-1}^k \ , \qquad i = \delta - \mu + 2, \ldots, n+1 \ . \tag{۲–۱۳}$$

به علاوه برای سایر توابع Bـاسپلاین داریم

$$B_i^k = \alpha_{i,i}\, \tilde{B}_i^k \ + \ \alpha_{i,i+1}\, \tilde{B}_{i+1}^k \ , \qquad i = \delta - k + 1, \ldots, \delta - \mu \ . \tag{۲–۱۴}$$

برهان. چون تعریف هر B_i^k تنها به زیردنبالهٔ $u_i, \ldots, u_{i+k}$ از دنبالهٔ گره‌ها وابسته است و چنین زیردنباله‌هایی به ازای $i \le \delta - k$، با درج t تغییر نمی‌کنند. بنابراین برای $i = 1, \ldots, \delta - k$ داریم $\tilde{B}_i^k = B_i^k$. با درج t در $\mathcal{U}$، t در مکان $\delta + 1$ قرار می‌گیرد و مطابق (۲–۸) زیردنباله‌های $u_i, \ldots, u_{i+k}$ به ازای $i \ge \delta + 1$ یک اندیس به جلو منتقل می‌شوند، امّا طبق (۲–۷) داریم $t = u_\delta = \cdots = u_{\delta-\mu+1}$ و بنابراین مقادیر گره‌ها در زیردنباله‌های $u_i, \ldots, u_{i+k}$ به ازای $i \ge \delta - \mu + 1$، برابر مقادیر گره‌ها در زیردنباله‌های $\tilde{u}_{i+1}, \ldots, \tilde{u}_{i+k+1}$ خواهد بود. بدین ترتیب برای $i = \delta - \mu + 2, \ldots, n+1$ داریم $\tilde{B}_i^k = B_{i-1}^k$.

تا اینجا نشان دادیم که مگر با یک انتقال اندیس، توابع Bـاسپلاین B_i^k به ازای $i \le \delta - k \ , \ i \ge \delta - \mu + 1$، بدون تغییر به پایهٔ جدید وارد می‌شوند.

حال برای $i = \delta - k + 1, \ldots, \delta - \mu$ داریم

$$B_i^k = \sum_{j=1}^{n+1} \alpha_{i,j}\, \tilde{B}_j^k \ .$$

برای مشخص کردن ضرایب $\alpha_{i,j}$، اتحاد فوق را به ازای برخی مقادیر پارامتر مورد ارزیابی قرار می‌دهیم. با ثابت نگهداشتن i، دو دنبالهٔ $s_1^{[2]}, s_2^{[2]}, \ldots, s_{n-i}^{[2]}$ و $s_1^{[1]}, s_2^{[1]}, \ldots, s_{i-1}^{[1]}$ از مقادیر پارامتر را به گونه‌ای انتخاب می‌کنیم که

$$s_j^{[1]} \in [u_j \; u_{j+1}] \,, \qquad \tilde{B}_{j+1}^k(s_j^{[1]}) = 0 \,, \quad j = 1, \ldots, (i-1) \,,$$

$$s_j^{[2]} \in [u_{i+k+j-1} \; u_{i+k+j}] \,, \qquad \tilde{B}_{i+j}^k(s_j^{[2]}) = 0 \,, \quad j = 1, \ldots, (n-i) \,.$$

در حالتی که دنبالهٔ گره‌های $\mathcal{U}$ حاوی هیچ گره تکراری نباشد، این انتخاب به سادگی به وسیلهٔ قراردادن $s_1^{[1]} = u_2, \ldots, s_{i-1}^{[1]} = u_i$ و $s_1^{[2]} = u_{i+k}, \ldots, s_{n-i}^{[2]} = u_{n+k-1}$ بدست می‌آید. با کمی دقت می‌توان نشان داد که در حالت کلی نیز چنین دنباله‌هایی را می‌توان تعیین کرد. ویژگی این دو دنباله چنین است که

$$B_i^k(s_j^{[1]}) = B_i^k(s_j^{[2]}) = 0 \,;$$

$$\tilde{B}_r^k(s_j^{[1]}) = 0 \,, \qquad r > j \quad \text{فقط وقتی}$$

$$\tilde{B}_r^k(s_j^{[2]}) = 0 \,, \qquad r < i + j + 1 \quad \text{فقط وقتی}$$

بدین ترتیب ملاحظه می‌شود که

$$\begin{aligned}
0 &= \alpha_{i,1} \, \tilde{B}_1^k(s_1^{[1]}) \\
0 &= \alpha_{i,1} \, \tilde{B}_1^k(s_2^{[1]}) + \alpha_{i,2} \, \tilde{B}_2^k(s_2^{[1]}) \\
&\vdots \\
0 &= \alpha_{i,1} \, \tilde{B}_1^k(s_{i-1}^{[1]}) + \cdots + \alpha_{i,i-1} \, \tilde{B}_{i-1}^k(s_{i-1}^{[1]})
\end{aligned}$$

و

$$\begin{aligned}
0 &= \alpha_{i,n+1} \, \tilde{B}_{n+1}^k(s_{n-i}^{[2]}) \\
0 &= \alpha_{i,n} \, \tilde{B}_n^k(s_{n-i-1}^{[2]}) + \alpha_{i,n+1} \, \tilde{B}_{n+1}^k(s_{n-i-1}^{[2]}) \\
&\vdots \\
0 &= \alpha_{i,i+2} \, \tilde{B}_{i+2}^k(s_1^{[2]}) + \cdots + \alpha_{i,n+1} \, \tilde{B}_{n+1}^k(s_1^{[2]}) \,.
\end{aligned}$$

بنابراین برای $j = ۱, \ldots, (i-۱),\ (i+۲), \ldots, (n+۱)$ بدست می‌آوریم $\alpha_{i,j} = ۰$. □

لم فوق نشان داد که هر تابع پایهٔ Bـاسپلاین در پایهٔ قدیم حداکثر به دو تابع پایهٔ Bـاسپلاین در پایهٔ جدید وابسته است، به طوری که ماتریس A_L تنها روی دو قطر اصلی خود دارای ضرایب غیرصفر است. قضیهٔ زیر این ضرایب را به طور دقیق تعیین می‌کند.

قضیه‌ی ۳.۲ فرض کنید B_i^k و $\tilde{B}_i^k$ به ترتیب توابع پایه Bـاسپلاین روی دنبالهٔ گره‌های $\mathcal{U}$ و $\tilde{\mathcal{U}} = \mathcal{U} \cup \{t\}$ باشند و μ, δ مطابق (۲–۶) و (۲–۷) تعریف شده‌اند، آنگاه داریم

$$B_i^k = \begin{cases} \tilde{B}_i^k\ , & ۱ \le i \le \delta - k\ , \\ w_i\, \tilde{B}_i^k\ +\ (۱ - w_{i+۱})\, \tilde{B}_{i+۱}^k\ , & \delta - k + ۱ \le i \le \delta - \mu\ , \\ \tilde{B}_{i+۱}^k\ , & \delta - \mu + ۱ \le i \le n \end{cases} \qquad (۱۵–۲)$$

که در آن ضرایب w_i به صورت

$$w_{\delta-\mu+۱} = ۰\ , \qquad \text{و} \qquad w_{\delta-k+۱} = ۱ \qquad (۱۶–۲)$$

$$w_i = \frac{t - u_i}{u_{i+k-۱} - u_i}\ , \qquad \delta - k + ۲ \le i \le \delta - \mu \qquad (۱۷–۲)$$

تعریف می‌شوند.

برهان. بنابر لم ۲.۲ کافیست ثابت کنیم $\alpha_{i,i} = w_i$ و $\alpha_{i,i+۱} = ۱ - w_{i+۱}$. برای بدست آوردن این دو ضریب، اتحاد (۲–۱۴) را به ازای برخی مقادیر پارامتر ارزیابی می‌کنیم. با هر انتخاب $s^{[۱]} \in [\tilde{u}_i\ \tilde{u}_{i+۱})$ و $s^{[۲]} \in [\tilde{u}_{i+k}\ \tilde{u}_{i+k+۱})$ به طور دلخواه، داریم $\tilde{B}_i^k(s^{[۲]}) = ۰, \tilde{B}_{i+۱}^k(s^{[۱]}) = ۰$ و بنابراین

$$\begin{aligned} B_i^k(s^{[۱]}) &= \alpha_{i,i}\ \tilde{B}_i^k(s^{[۱]}) \\ B_i^k(s^{[۲]}) &= \alpha_{i,i+۱}\ \tilde{B}_{i+۱}^k(s^{[۲]}) \end{aligned}$$

و با استفاده از لم ۱.۲ برای $s^{[۱]}$ داریم

$$\begin{aligned} \alpha_{i,i} &= \frac{B_i^k(s^{[۱]})}{\tilde{B}_i^k(s^{[۱]})} \\ &= \frac{(u_{i+k-۲} - u_i) \cdots (t - u_i) \cdots (u_{i+۱} - u_i)}{(u_{i+k-۱} - u_i) \cdots (u_{i+۱} - u_i)} \qquad \text{وقتی } i > \delta - k + ۱ \\ &= \frac{t - u_i}{u_{i+k-۱} - u_i} = w_i, \end{aligned}$$

و وقتی $i = \delta - k + ۱$،

$$\begin{aligned} &= \frac{(u_{i+k-۱} - u_i) \cdots (u_{i+۱} - u_i)}{(u_{i+k-۱} - u_i) \cdots (u_{i+۱} - u_i)} \\ &= ۱ = w_{\delta-k+۱} \ . \end{aligned}$$

به همین ترتیب برای $s^{[۲]}$،

$$\begin{aligned} \alpha_{i,i+۱} &= \frac{B_i^k(s^{[۲]})}{\tilde{B}_{i+۱}^k(s^{[۲]})} \\ &= \frac{(u_{i+k} - u_{i+۲}) \cdots (u_{i+k} - t) \cdots (u_{i+k} - u_{i+k-۱})}{(u_{i+k} - u_{i+۱}) \cdots (u_{i+k} - u_{i+k-۱})} \qquad \text{وقتی } i < \delta - \mu + ۱ \\ &= \frac{u_{i+k} - t}{u_{i+k} - u_{i+۱}} = ۱ - w_{i+۱}, \end{aligned}$$

و وقتی $i = \delta - \mu + ۱$،

$$\begin{aligned} &= \frac{(u_{i+k} - u_{i+۱}) \cdots (u_{i+k} - u_{i+k-۱})}{(u_{i+k} - u_{i+۱}) \cdots (u_{i+k} - u_{i+k-۱})} \\ &= ۱ = ۱ - w_{i+۱} \ . \end{aligned}$$

□

بدین ترتیب می‌توان ماتریس تبدیل پایهٔ A_L را به صورت

$$A_L = \begin{pmatrix} I_{\delta-k} & ۰ & ۰ \\ ۰ & A & ۰ \\ ۰ & ۰ & I_{n-\delta+\mu} \end{pmatrix}_{(n+۱)\times n} \qquad (۱۸-۲)$$

نوشت که I_M نشاندهندهٔ ماتریس همانی $M \times M$ است. A **ماتریس کوچک درج گره** نامیده می‌شود و به صورت

$$A = \begin{pmatrix} ۱ = w_{\delta-k+۱} & ۰ & ۰ & \cdots & ۰ \\ ۱ - w_{\delta-k+۲} & w_{\delta-k+۲} & ۰ & \cdots & ۰ \\ ۰ & ۱ - w_{\delta-k+۳} & w_{\delta-k+۳} & \cdots & ۰ \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ ۰ & \cdots & ۰ & ۱ - w_{\delta-\mu} & w_{\delta-\mu} \\ ۰ & \cdots & & ۰ & ۱ - w_{\delta-\mu+۱} = ۱ \end{pmatrix}_{(k-\mu+۱)\times(k-\mu)} \qquad (۱۹-۲)$$

تعریف می‌شود.

اثر ماتریس A_L برروی چندضلعی کنترلی $\{c_i\}_{i=۱}^n$، الگوریتم درج تک گره را بدست می‌دهد.

۲—۲—۱ الگوریتم درج تک گره

فرض کنید $\{c_i\}_{i=1}^{n}$ چندضلعی کنترلی منحنی Bـاسپلاین Q روی دنبالهٔ گره‌های $\mathcal{U}$ باشد، آنگاه برای منحنی Bـاسپلاین $\tilde{Q}$ روی دنبالهٔ گره‌های $\tilde{\mathcal{U}}$ با چندضلعی کنترلی $\{\tilde{c}_i\}_{i=1}^{n+1}$ که از رابطهٔ

$$\tilde{c}_i \;=\; \begin{cases} c_i & 1 \le i \le \delta - k + 1 \\ w_i\, c_i \;+\; (1 - w_i)\, c_{i-1} & \delta - k + 2 \le i \le \delta - \mu \\ c_{i-1} & \delta - \mu + 1 \le i \le n + 1 \end{cases} \qquad (2\text{—}20)$$

تعیین می‌شود، داریم $Q \equiv \tilde{Q}$.

این رابطه را می‌توان در فرم ماتریسی به صورت $\tilde{\mathbf{c}}_L = A_L \mathbf{c}_L$ نوشت. از آنجا که در درج یک گره، بسیاری از رئوس کنترلی تغییر نمی‌کنند، بکارگیری A_L به این صورت منطقی به نظر نمی‌رسد. در عوض ماتریس A را تنها برای رئوس کنترلی که تغییر می‌کنند به کار می‌بریم. اگر قرار دهیم

$$\mathbf{c} \;=\; \begin{pmatrix} c_{\delta-k+1} \\ c_{\delta-k+2} \\ \vdots \\ c_{\delta-\mu} \end{pmatrix} \quad , \quad \tilde{\mathbf{c}} \;=\; \begin{pmatrix} \tilde{c}_{\delta-k+1} \\ c_{\delta-k+2} \\ \vdots \\ c_{\delta-\mu} \\ \tilde{c}_{\delta-\mu+1} \end{pmatrix} \qquad (2\text{—}21)$$

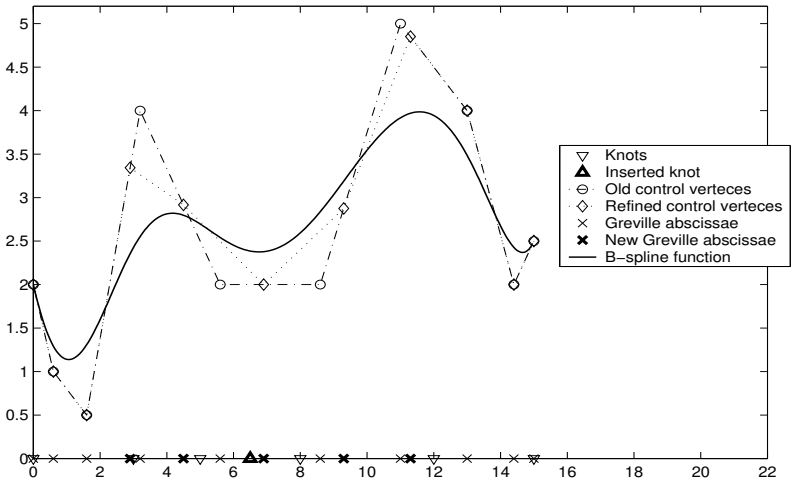
آنگاه داریم

$$\tilde{\mathbf{c}} \;=\; A\, \mathbf{c}\,. \qquad (2\text{—}22)$$

در پیاده‌سازی این الگوریتم به سادگی می‌توان ترتیبی داد که عمل تظریف روی خود $\mathbf{c}$ صورت گیرد و نیازی به تخصیص حافظه به جز یک واحد برای $\tilde{\mathbf{c}}$ نباشد.

این فرایند یک تعبیر جالب هندسی دارد که باز هم توسط چندضلعی دِبور و نقاط گرویل ξ_i بیان می‌شود (به شکل ۲—۱ نگاه کنید). اگر $\{\tilde{\xi}_i\}_{i=1}^{n+1}$ نشاندهندهٔ نقاط گرویل برای دنبالهٔ گره‌های $\tilde{\mathcal{U}}$ باشد آنگاه با یک عبارت ساده می‌توان گفت: مقادیر $\tilde{c}_i$ از درونیابی چندضلعی دِبور $\left\{ \begin{pmatrix} \xi_j \\ c_j \end{pmatrix} \right\}_{j=1}^{n}$ روی نقاط $\tilde{\xi}_i$ بدست می‌آیند:

$$\tilde{c}_i \;=\; \frac{\tilde{\xi}_i - \xi_i}{\xi_{i+1} - \xi_i}\,(c_{i+1} - c_i) \;+\; c_i \qquad i = 1, 2, \ldots, n+1\,.$$



شکل ۲—۱: درج گره در یک منحنی Bـاسپلاین مکعبی.

دربارهٔ ضرایب w_i در بخش ۳—۱—۲ بیشتر صحبت می‌کنیم و نشان می‌دهیم که $0 \leq w_i \leq 1$، یعنی عمل فوق حقیقتاً یک درونیابی است.

۲—۲—۲ درج مکرر یک گره

حالت خاص دیگری که در الگوریتمهای درج گره قابل توجه است، درج کردن یک گره به طور مکرر تا جایی است که مرتبهٔ تکرار آن گره در $\mathcal{U}$ از حداکثر مقدار مجاز، یعنی k تجاوز نکند. این روش درج علاوه بر اینکه همانند سایر روشهای درج گره باعث تظریف منحنی می‌شود، دو کاربردهای مهم دیگر نیز دارد. اول آنکه منحنی در گره u_i با تکرار k، رأس کنترلی c_{i-k+1} را درونیابی می‌کند و از این خاصیت می‌توان برای پارامتری کردن منحنی استفاده کرد، بدین ترتیب که پارامتر t را می‌گیریم و آن را به عنوان گره جدید، k بار در $\mathcal{U}$ درج می‌کنیم. کاربرد دوم تبدیل یک منحنی Bـاسپلاین به یک دنبالهٔ پیوسته از منحنیهای بزیه است. طبق قضیهٔ ۴.۱، اگر مرتبهٔ تکرار گره‌های درونی $\mathcal{U}$ را تا $k-1$ و دو گره ابتدایی و انتهایی را تا k افزایش دهیم، چندضلعی کنترلی منحنی Bـاسپلاین حاصل، شامل یک دنبالهٔ پیوسته از چندضلعی‌های کنترلی بزیه است.

پس به طور کلی فرض کنید می‌خواهیم گره t را $r \leq k-\mu$ بار در $\mathcal{U}$ درج کنیم. اگر به (۲—۸)

توجه کنید، می‌بینید که پس از هر بار درج t، μ یک واحد افزایش می‌یابد و بدین ترتیب ماتریس A و رئوس کنترلی که تغییر می‌کنند، $\mathbf{c}$، ابعاد کوچکتری پیدا می‌کنند. بنابراین منطقی است که به جای r بار تکرار الگوریتم درج تک گره، این الگوریتم را به طور مستقیم و خلاصه‌تری طراحی کنیم. با ترکیب متوالی روابط (۲–۸) و (۲–۲۰) الگوریتم درج مکرر یک گره به صورت زیر بدست خواهد آمد.

۱) رئوس کنترلی $\{c_i\}_{i=1}^{n}$، گره t و عدد $r \le k-\mu$ داده شده‌اند. می‌خواهیم گره t را r بار در $\mathcal{U}$ درج کنیم.

۲) قراربده $j=0$ و برای $i=\delta-k+1,\ldots,\delta-\mu$ قرار بده $c_i^{[0]} := c_i$.

۳) برای $j=1,\ldots,r$ تکرار کن:

$$c_i^{[j]} \;=\; w_i^j\, c_i^{[j-1]} \;+\; (1-w_i^j)\, c_{i-1}^{[j-1]}, \qquad\qquad j+\delta-k+1 \le i \le \delta-\mu$$

به طوری که

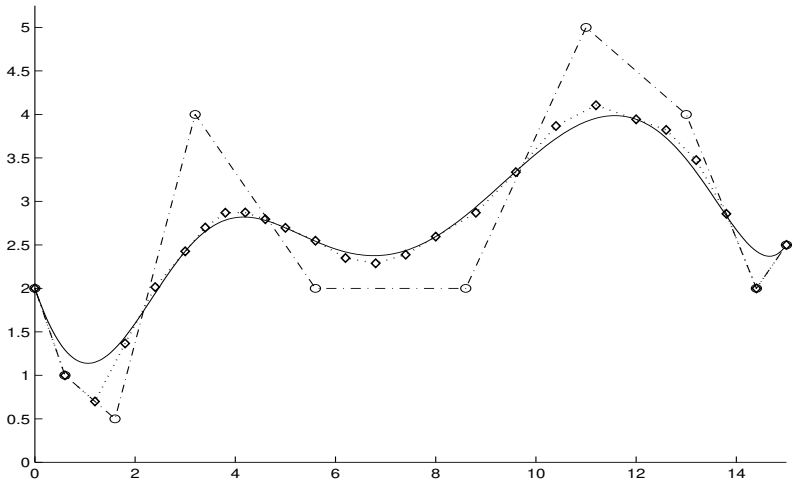
$$w_i^j \;=\; \frac{t-u_i}{u_{i+k-j}-u_i}\;.$$

۴) آنگاه رئوس کنترلی جدید $\{\tilde{c}_i\}_{i=1}^{m}$ به وسیلهٔ

$$\tilde{c}_i \;=\; \begin{cases} c_i & 1 \le i \le \delta-k+1 \\ c_i^{[i-\delta+k-1]} & \delta-k+2 \le i \le \delta-k+1+r \\ c_i^{[r]} & \delta-k+2+r \le i \le \delta-\mu \\ c_{\delta-\mu}^{[r-i+\delta-\mu]} & \delta-\mu+1 \le i \le \delta-\mu+r-1 \\ c_{i-r} & \delta-\mu+r \le i \le n+r=m \end{cases}$$

تعیین می‌شوند.

در شکل ۲–۲، مرتبهٔ تکرار هر یک از گره‌های یک منحنی Bـاسپلاین به وسیلهٔ الگوریتم فوق تا $k-1=3$ بالا برده می‌شود، و بدین ترتیب یک دنبالهٔ پیوسته از چندضلعیهای کنترلی بزیه برای قطعات همین منحنی بدست می آید.



شکل ۲—۲: درج مکرر تمام گره‌های اولیه یک منحنی Bـاسپلاین مکعبی تا رسیدن مرتبه تکرار هر یک از آنها به ۳: چندضلعی کنترلی اولیه (خط—نقطه)، چندضلعی کنترلی تظریف شده (نقطه‌چین).

۲—۳ درج چند گره

هرچند با بکارگیریهای متوالی الگوریتم درج تک گره می‌توان گره‌های متعددی را نیز در دنباله گره‌ها درج کرد، همانند بحث درج مکرر یک گره، به نظر می‌رسد که این کار با خلاصه کردن برخی تغییرات متوالی روی متغیرها، به طور مستقیم و با سرعت بیشتری، انجام پذیر است.

در اینجا بدون ذکر جزئیات و اثباتها، به طور خلاصه الگوریتم درج همزمان چند گره، که به الگوریتم اسلو[۱] معروف است، بیان می‌شود. این روش را ابتدا **کوهن، لیخ و ریزنفلد** در [۱۸] ارائه کردند. در ادامه این الگوریتم به نقل از [۳۳] بیان می‌شود.

برای درج همزمان چند گره در حالت کلی، فرض کنید l گره $t_1, \ldots, t_l$ را به طور همزمان در $\mathcal{U}$ درج کرده‌ایم. بدین ترتیب داریم

$$\begin{aligned} \tilde{\mathcal{U}} &= \mathcal{U} \cup \{t_1, t_2, \ldots, t_l\} \\ m &= n + l\,. \end{aligned}$$

اگر قرار دهیم

$$\mathbf{B}^k = [B_1^k \ \cdots \ B_n^k]^T, \qquad \tilde{\mathbf{B}}^k = [\tilde{B}_1^k \ \cdots \ \tilde{B}_m^k]^T$$

[۱] Oslo Algorithm

آنگاه دنباله‌ای از ماتریسهای تبدیل $\{A^{[r]}\}_{r=1}^{k}$ وجود دارد به طوری که

$$\mathbf{B}^r = A^{[r]T}\tilde{\mathbf{B}}^r , \qquad 1 \le r \le k . \qquad (۲۳-۲)$$

$A^{[k]}$ ماتریس تبدیل پایهٔ مرتبهٔ k با ابعاد $n \times m$ است و ضرایب این ماتریس $\alpha_{i,j}^{[k]}$ به صورت بازگشتی با رابطهٔ زیر تعیین می‌شود:

$$\alpha_{i,j}^{[1]} = \begin{cases} 1 & \tilde{u}_j \in [u_i \; u_{i+1}) \\ 0 & \text{در غیر اینصورت} \end{cases}$$

وقتی $k = 1$ و

$$\alpha_{i,j}^{[k]} = \frac{\tilde{u}_{j+k-1} - u_i}{u_{i+k-1} - u_i}\,\alpha_{i,j}^{[k-1]} + \frac{u_{i+k} - \tilde{u}_{j+k-1}}{u_{i+k} - u_{i+1}}\,\alpha_{i+1,j}^{[k-1]}$$

وقتی $k > 1$. اگر این روابطه را با روابطهٔ (۱-۱) و (۱-۲) برای تعریف توابع پایهٔ Bـاسپلاین مقایسه کنید، متوجه شباهت آنها می‌شوید. از این جهت است که این ضرایب $\alpha_{i,j}^{[k]}$ در برخی منابع به عنوان Bـاسپلاینهای گسسته شناخته می‌شوند. عبارت گسسته باین خاطر گفته می‌شود که در تعریف $\alpha_{i,j}^{[k]}$ عدد صحیح j به نوعی به جای پارامتر پیوستهٔ t در $B_i^k(t)$ قرار گرفته است.

تمام خواص توابع پایهٔ Bـاسپلاین، به صورتی برای Bـاسپلاینهای گسسته نیز وجود دارند:

- نامنفی بودن $A^{[k]}$. برای هر i, j, k داریم $\alpha_{i,j}^{[k]} \ge 0$.
- حاصلجمع ستونی واحد. برای هر j, k داریم $\sum_{i=1}^{n} \alpha_{i,j}^{[k]} = 1$.
- پرصفر[۲] بودن $A^{[k]}$. برای هر j, k وقتی $i < \delta(\tilde{u}_i) - k + 1$ یا $i > \delta(\tilde{u}_i)$ داریم $\alpha_{i,j}^{[k]} = 0$. به خصوص چون $\tilde{u}_j \le \tilde{u}_{j+1}$ می‌توان گفت که $A^{[k]}$ یک ماتریس نواری است.

با این حال در عمل برای محاسبهٔ چندضلعی کنترلی جدید $\{\tilde{c}_i\}_{i=1}^{m}$، به جای استفاده صریح از ماتریس تبدیل پایه $A^{[k]}$ و رابطهٔ $\tilde{\mathbf{c}} = A^{[k]}\mathbf{c}$، از یک الگوریتم بازگشتی مشابه (۱-۱۶) ، استفاده می‌شود [۳۳].

این الگوریتم، با تنظیم الگوریتم اسلو برای محاسبهٔ مستقیم رئوس کنترلی تظریف شده، به صورت زیر بدست می آید.

---

[۲] Sparse

۱) رئوس کنترلی $\{c_i\}_{i=۱}^{n}$ و دنبالهٔ گره‌های $\mathcal{U} \subset \tilde{\mathcal{U}}$ داده شده است. می‌خواهیم رئوس کنترلی $\{\tilde{c}_j\}_{j=۱}^{m}$ را طوری بدست آوریم که $\mathbf{B}_L^T \mathbf{c}_L = \tilde{\mathbf{B}}_L^T \tilde{\mathbf{c}}_L$.

۲) برای هر $j = ۱, ۲, \ldots, m$ محاسبات زیر را انجام بده.

(a) قرار بده $r = k$ و $\delta = \delta(\tilde{u}_j)$ و برای $i = \delta - k + ۱, \ldots, \delta$ قرار بده $c_i^{[k]} := c_i$

(b) برای $r = k - ۱, \ldots, ۱$ تکرار کن:

$$c_i^{[r]} = \frac{(\tilde{u}_{j+r} - u_i)\, c_i^{[r+۱]} + (u_{i+r} - \tilde{u}_{j+r})\, c_{i-۱}^{[r+۱]}}{u_{i+r} - u_i}, \qquad i = \delta - r + ۱ \ldots, \delta\,.$$

۳) داریم $\tilde{c}_j = c_\delta^{[۱]}$.

فصل ۳

حذف گره

۳—۱ زمینه

در این فصل به طور مفصل به مسألهٔ حذف گره از دیدگاه نظری پرداخته می‌شود. در فصل بعدی جزئیات لازم برای پیاده‌سازی الگوریتمهایی برای این مسأله ارائه می‌شود و در پایان به برخی از کاربردهای روشهای حذف و درج گره اشاره می‌شود. در این رساله اساساً مسألهٔ حذف یک گره از دنبالهٔ گره‌های یک منحنی Bـاسپلاین مد نظر است.

تا پایان این مبحث قراردادهای زیر را مفروض می‌گیریم.

چون روش حذف گره به نوعی معکوس‌سازی روش درج گره است و نیز برای استفاده از روابط بدست آمده در بخش ۲—۲، بار دیگر نمادهای به کارگرفته شده در آنجا را در نظر می‌گیریم.

اگر t گره درج شده در $\mathcal{U}$ باشد و $\tilde{\mathcal{U}} = \mathcal{U} \cup \{t\}$، آنگاه اصطلاحاً فضای $\mathcal{S}^k_{n,\mathcal{U}}$ را فضای درشت[۱] و $\mathcal{S}^k_{n+1,\tilde{\mathcal{U}}}$ را فضای ظریف[۲] می‌گوییم. $\mathcal{B} = \mathcal{B}^k_{n,\mathcal{U}}$ پایهٔ Bـاسپلاین درشت و $\tilde{\mathcal{B}} = \mathcal{B}^k_{n+1,\tilde{\mathcal{U}}}$ پایهٔ Bـاسپلاین ظریف هستند. بدین ترتیب $\tilde{u}_{\tilde{\delta}} = t$ که $\tilde{\delta} = \delta + 1$ و مرتبهٔ تکرار آن در دنبالهٔ گره‌های ظریفتر به صورت $\tilde{\mu} = \mu + 1$ است. رئوس کنترلی که در فرایند درج (همچنین حذف) تغییر می‌کنند، در پایهٔ درشت و پایهٔ

[۱] coarse

[۲] fine

ظریف به ترتیب با بردارهای

$$\mathbf{c} = [c_{\bar{\delta}-k} \cdots c_{\bar{\delta}-\bar{\mu}}]^T \quad , \quad \tilde{\mathbf{c}} = [\tilde{c}_{\bar{\delta}-k} \cdots \tilde{c}_{\bar{\delta}-\bar{\mu}} \tilde{c}_{\bar{\delta}-\bar{\mu}+1}]^T \qquad (۱-۳)$$

نشان داده می‌شوند. همچنین توابع پایه‌ای که حین این فرایند تغییر می‌یابند، در پایهٔ درشت و پایهٔ ظریف به ترتیب با بردارهای

$$\mathbf{B} = [B^k_{\bar{\delta}-k} \cdots B^k_{\bar{\delta}-\bar{\mu}}]^T \quad , \quad \tilde{\mathbf{B}} = [\tilde{B}^k_{\bar{\delta}-k} \cdots \tilde{B}^k_{\bar{\delta}-\bar{\mu}} \tilde{B}^k_{\bar{\delta}-\bar{\mu}+1}]^T \qquad (۲-۳)$$

نشان داده می‌شوند. بدین ترتیب روابطهٔ درج گره با توجه به روابط (۲−۱۹) و (۲−۲۲) به صورت

$$\mathbf{B} = A^T \tilde{\mathbf{B}} \qquad (۳-۳)$$

$$\tilde{\mathbf{c}} = A\mathbf{c} \qquad (۴-۳)$$

در می‌آیند، که A ماتریس کوچک یا موضعی درج گره مطابق (۲−۱۹) است و با نمادگذاری جدید دارای ابعاد $(k - \tilde{\mu} + ۲) \times (k - \tilde{\mu} + ۱)$ است.

هدف از حذف گره، تعیین *بهترین* چندضلعی کنترلی با تعداد کمتری از رئوس کنترلی است به طوری که منحنی Bـاسپلاین حاصل از آن دارای کمترین اختلاف با منحنی اولیهٔ قبل از حذف، باشد. کلمهٔ بهترین به این خاطر گفته می‌شود که در حالت کلی، فرایند حذف به طور یکتا انجام نمی‌شود. برای درک چگونگی عمل حذف یک گره حالتی را در نظر بگیرید که ابتدا روی یک منحنی، عمل درج گره را انجام می‌دهیم. حال منطقی است که اگر بخواهیم همان گره را از منحنی تظریف شده حذف کنیم، دقیقاً همان منحنی اولیه بدست آید. این کار اساساً با معکوس سازی رابطهٔ (۳−۴) ، برای بدست آوردن $\mathbf{c}$ برحسب $\tilde{\mathbf{c}}$ به سادگی انجام می‌شود. در حالت کلی ماتریس درج گره A، یک سطر بیش از رتبه‌اش دارد و ناسازگار است، پس معکوس آن به طور یکتا تعیین نمی‌شود. امّا در این حالت خاص A سازگار است و می‌توان با صرفنظر کردن از هر یک از سطرهای آن، یک معکوس برای آن محاسبه کرد و اطمینان داشت که تنها جواب بدست آمده، بر منحنی اولیه منطبق است.

در حالت کلی این نحوهٔ خاص معکوس سازی که آنرا حذف گره مجازی[۳] می‌نامیم، مرحلهٔ اول در یک الگوریتم حذف گره است. با افزودن یک متغیر مجهول دیگر مثل α به دستگاه ناسازگار (۳−۴) به

[۳] Pseudo-knot removal

صورت $A\mathbf{c} = \tilde{\mathbf{c}} + \alpha\mathbf{C}$ که $\mathbf{C}$ یک بردار ثابت و متناسب با نوع روش حذف گره است، فضای جواب این دستگاه را به صورت یک ترکیب گرانیگاهی روی دو جواب خاص بدست آمده از روش حذف گره مجازی فرمولبندی می‌کنیم. در مرحلهٔ دوم با هدف مینیمم سازی خطای بین منحنی اولیه و منحنی حاصل از حذف گره، ضرایب وزنی مناسب در این ترکیب گرانیگاهی را محاسبه می‌کنیم. این مرحله را حذف گره بهترین تقریب می‌نامیم.

قبل از ادامهٔ بحث و برای روشنتر شدن مطلب، یک روش کاهش درجه[۴] در منحنی بزیه را که دارای چارچوب مشابه با روشهای حذف گره در Bـاسپلاینهاست، به عنوان مثال ذکر می‌کنیم.

۳—۱—۱ کاهش درجه در منحنی بزیه

مطابق مطالبی که در بخش ۱—۴—۲ دربارهٔ بالا بردن درجه گفته شد، یک منحنی بزیه از درجهٔ n با رئوس کنترلی $c_0, c_1, \ldots, c_n$ را می‌توان به عنوان یک منحنی بزیهٔ درجهٔ $n+1$ با رئوس کنترلی

$$\tilde{c}_i = \frac{i}{n+1}\, c_{i-1} + (1 - \frac{i}{n+1})\, c_i\,, \qquad i = 0, 1, \ldots, (n+1) \qquad (۳—۵)$$

نمایش داد. اگر بخواهیم یک منحنی بزیهٔ درجهٔ $n+1$ را با رئوس کنترلی کمتری نمایش دهیم، ابتدا دو راه حل ساده به نظر می‌رسد؛ در راه اول رأس $c_0 = \tilde{c}_0$ را ثابت قرار می‌دهیم و رابطهٔ (۳—۵) را از اول به آخر برای بدست آوردن c_i حل می‌کنیم:

$$c_i^I = \frac{(n+1)\,\tilde{c}_i - i\, c_{i-1}^I}{n-i+1}\,, \qquad i = 0, 1, \ldots, n$$

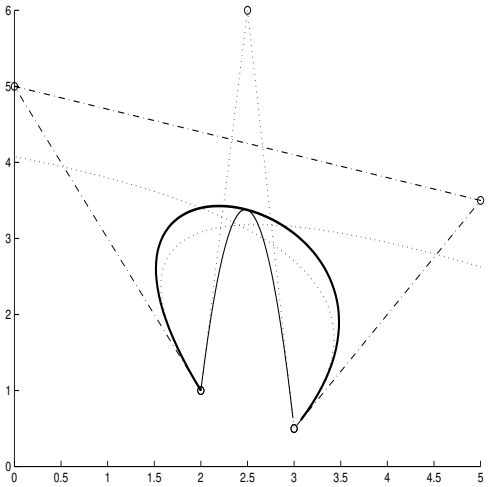
و در راه دوم رأس $c_n = \tilde{c}_{n+1}$ را ثابت می‌گیریم و رابطهٔ (۳—۵) را از آخر به اول برای c_{i-1} حل می‌کنیم:

$$c_{i-1}^{II} = \frac{(n+1)\,\tilde{c}_i - (n-i+1)\, c_i^{II}}{i}\,, \qquad i = n+1, n, \ldots, 1\,.$$

مسلماً در حالت کلی، به جز در مواردی که رئوس کنترلی $\tilde{c}_i$ حقیقتاً برای مدل کردن یک منحنی درجهٔ $n+1$ اضافی هستند، ضرایب c_i^I و c_i^{II} دارای مقادیر یکسانی با c_i نخواهند بود.

[۴] Degree reduction

به همین خاطر معمولاً یک ترکیب گرانیگاهی از آنها به صورت $\bar{c}_i = (1 - \lambda_i)\, c_i^I + \lambda_i\, c_i^{II}$ انتخاب می‌شود که در آن $0 \leq \lambda_i \leq 1$. مثلاً در تقریب بزیه درجه ۳ (مکعبی) به بزیه درجهٔ ۲ (سهمی)، انتخاب $\lambda_0 = 0, \lambda_1 = 1/2, \lambda_2 = 1$ یک تقریب خوب بدست می‌دهد. در شکل ۳—۱ یک نمونه از این روش کاهش درجه نشان داده شده‌است. این روش کاهش درجه آنطورکه در [۲۱] نقل شده است، در اینجا آورده شد. دربارهٔ موضوع کاهش درجه در منحنی بزیه، به همین میزان اکتفا می‌کنیم و از پرداختن عمیق‌تر به آن خودداری می‌کنیم. برای جزئیات بیشتر می‌توانید به [۴۱]، [۲۲] و [۱۹] رجوع کنید.



شکل ۳—۱: یک روش کاهش درجه برای تبدیل منحنی بزیه درجه سه به درجه دو: منحنی بزیه اولیه (منحنی کلفت)، منحنی بزیه حاصل از کاهش درجه (منحنی نازک)، بخشی از دو منحنی متناظر با $\mathbf{c}^I$ و $\mathbf{c}^{II}$ به صورت نقطه‌چین در شکل قرار گرفته‌است.

ادامهٔ مطالب این فصل به جز چند لم و تعریف از [۲۰] نقل می‌شود.

۳—۱—۲ ضرایب براکتی

مجدداً به بحث اصلی خود بازمی‌گردیم. امّا قبل از پرداختن به الگوریتم حذف گره، برخی نمادها را که در الگوریتمها و برخی از اثباتها بکار می‌آیند، معرفی می‌کنیم. برای اختصار در اندیسگذاریها، قرار می‌دهیم $n_1 = \tilde{\delta} - k$, $n_2 = \tilde{\delta} - \tilde{\mu} + 1$, $J = \tilde{\delta} - k - \tilde{\mu} + 1$ و یک بار دیگر ضرایب ماتریس (کوچک) درج گره

A را به دقت بررسی می‌کنیم. طبق (۲–۱۹) داریم

$$A = \begin{pmatrix} ۱ = w_{n_۱} & ۰ & ۰ & \cdots & ۰ \\ ۱ - w_{n_۱+۱} & w_{n_۱+۱} & ۰ & \cdots & ۰ \\ ۰ & ۱ - w_{n_۱+۲} & w_{n_۱+۲} & \cdots & ۰ \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ ۰ & \cdots & ۰ & ۱ - w_{n_۲-۱} & w_{n_۲-۱} \\ ۰ & \cdots & & ۰ & ۱ - w_{n_۲} = ۱ \end{pmatrix} \qquad (۶–۳)$$

که w_i ها را با توجه به (۲–۸) و (۲–۱۶) و (۲–۱۷) می‌توان برحسب گره‌های $\tilde{\mathcal{U}}$ به صورت

$$w_i = \frac{\tilde{u}_{\bar{\delta}} - \tilde{u}_i}{\tilde{u}_{i+k} - \tilde{u}_i}\,, \qquad i = n_۱, \ldots, n_۲ \qquad (۷–۳)$$

بازنویسی کرد.

لم ۱.۳ برای ضرایب w_i تعریف شده در (۳–۷) ، داریم

$$۱ = w_{n_۱} > w_{n_۱+۱} \geq \ldots \geq w_{n_۲-۱} > w_{n_۲} = ۰\,.$$

برهان. برای $n_۱ \leq i \leq n_۲ - ۱$ داریم

$$\tilde{u}_i \leq \tilde{u}_{i+۱} \;\Rightarrow\; \tilde{u}_{\bar{\delta}} - \tilde{u}_i \geq \tilde{u}_{\bar{\delta}} - \tilde{u}_{i+۱} \quad \text{و} \quad \tilde{u}_{i+k} \leq \tilde{u}_{i+k+۱} \;\Rightarrow\; \tilde{u}_{i+k+۱} - \tilde{u}_{\bar{\delta}} \geq \tilde{u}_{i+k} - \tilde{u}_{\bar{\delta}}\,.$$

به علاوه، با توجه به (۲–۶) و (۲–۷) ، نابرابری اول برای $i = n_۲ - ۱$ و نابرابری دوم برای $i = n_۱$ اکید خواهد بود. بنابراین

$$\begin{aligned} (\tilde{u}_{\bar{\delta}} - \tilde{u}_i)\,(\tilde{u}_{i+k+۱} - \tilde{u}_{i+۱}) &= (\tilde{u}_{\bar{\delta}} - \tilde{u}_i)\,(\tilde{u}_{i+k+۱} - \tilde{u}_{\bar{\delta}}) + (\tilde{u}_{\bar{\delta}} - \tilde{u}_i)\,(\tilde{u}_{\bar{\delta}} - \tilde{u}_{i+۱}) \\ &\geq (\tilde{u}_{\bar{\delta}} - \tilde{u}_{i+۱})\,(\tilde{u}_{i+k} - \tilde{u}_{\bar{\delta}}) + (\tilde{u}_{\bar{\delta}} - \tilde{u}_i)\,(\tilde{u}_{\bar{\delta}} - \tilde{u}_{i+۱}) \\ &= (\tilde{u}_{\bar{\delta}} - \tilde{u}_{i+۱})\,(\tilde{u}_{i+k} - \tilde{u}_i)\,. \end{aligned}$$

در نتیجه $w_i \geq w_{i+۱}$ و به خصوص برای $i = n_۲ - ۱$, $i = n_۱$ نابرابری اکید است. □

برای ادامهٔ بحث قرارداد زیر را تا پایان می‌پذیریم.

قرارداد. $m_۱$ و $m_۲$ دو عدد صحیح نامنفی هستند به طوری که $M := k - \tilde{\mu} + ۱ - m_۱ - m_۲ \geq ۰$.

لم بعدی یک ویژگی مهم A را که در مسائل تقریب به کار خواهد رفت، بیان می‌کند.

لم ۲.۳ اگر A_{m_1,m_2}، ماتریس با ابعاد $(M+1)\times M$ باشد که از حذف m_1 سطر و ستون اول و m_2 سطر و ستون آخر A بدست آمده باشد، آنگاه همهٔ کهادهای مرتبهٔ M آن مثبت هستند.

برهان. اگر $A_{m_1,m_2:j}$ نشاندهندهٔ ماتریس مربعی حاصل از حذف سطر j ام از ماتریس A_{m_1,m_2} باشد، آنگاه دترمینان آن دقیقاً برابر حاصلضرب تمام ضرایب روی قطرش است و بر اساس لم ۱.۳ حکم ثابت می‌شود:

$$\det A_{m_1,m_2:j} = \Big(\prod_{i=n_1+m_1}^{n_1+j-2} w_i\Big)\Big(\prod_{i=n_1+j}^{n_2-m_2}(1-w_i)\Big) > 0\ , \quad j = 1+m_1,\ldots,M+m_1\ . \quad (3-8)$$

□

تبصره. حکم مشابه برای ماتریس بزرگ A_L برقرار نیست. چون اکثر کهادهای مرتبه n آن صفر هستند.

با تعریف ضرایب براکتی[5] که به صورت $\begin{bmatrix} b \\ a \end{bmatrix}$ نوشته می‌شود، مطلب را ادامه می‌دهیم. از دیدگاه شهودی، این ضرایب با شبه وارون ماتریس A و دستگاه معادلات نرمال در ارتباط هستند و بسیاری از محاسبات بعدی به طور فشرده‌تری بر حسب آنها بیان می‌شود.

تعریف ۳.۳ برای اعداد صحیح a,b که $\tilde{\mu}-1 \le b \le k-1$ و $\tilde{\mu} \le a \le b+2$، مقدار مثبت

$$\begin{bmatrix} b \\ a \end{bmatrix} := \begin{cases} \frac{1}{w_{a+J-1}} \prod_{i=a+J}^{b+J} \frac{1-w_i}{w_i}\ , & a \le b\ , \\ \frac{1}{w_{a+J-1}}\ , & a = b+1\ , \\ \frac{1}{1-w_{a+J-1}}\ , & a = b+2 \end{cases} \quad (3-9)$$

ضریب براکتی a,b نامیده می‌شود.

لم بعدی دو رابطهٔ مفید برای محاسبهٔ این ضرایب ارائه می‌کند که بدون ذکر اثبات آورده می‌شود.

لم ۴.۳ ضرایب براکتی $\begin{bmatrix} b \\ a \end{bmatrix}$ در دو رابطهٔ بازگشتی زیر صدق می‌کنند:

$$\begin{bmatrix} b \\ a+1 \end{bmatrix} = \frac{w_{a+J-1}}{1-w_{a+J}} \begin{bmatrix} b \\ a \end{bmatrix} \qquad \tilde{\mu}-1 \le b \le k-1,\ \tilde{\mu} \le a \le b+1 \quad (3-10)$$

$$\begin{bmatrix} b+1 \\ a \end{bmatrix} = \frac{1-w_{b+J+1}}{w_{b+J+1}} \begin{bmatrix} b \\ a \end{bmatrix} \qquad \tilde{\mu}-1 \le b \le k-2,\ \tilde{\mu} \le a \le b+2 \quad (3-11)$$

[5] Bracket coefficients

آنچه در محاسبات مورد نیاز در این رساله، بسیار استفاده می‌شود، حالت خاصی از ضرایب براکتی با مؤلفهٔ بالایی $b = k - ۱$ است. این ضرایب را می توان با استفاده از رابطهٔ (۳–۱۰) به طور بازگشتی بدست آورد، بدین ترتیب که با $\begin{bmatrix} k-۱ \\ k+۱ \end{bmatrix} = ۱$ شروع می‌کنیم و با قرار دادن $j = k-۱, k-۲, \ldots, \tilde{\mu}-۱$ به ترتیب ضرایب $\begin{bmatrix} k-۱ \\ j+۱ \end{bmatrix}$ را بدست می آوریم.

در ادامه دو ویژگی مهم و جالب این ضرایب تعیین می‌شوند.

لم ۵.۳ برای هر دنبالهٔ گره‌های دلخواه $\mathcal{U}$ داریم

$$\sum_{j=\bar{\mu}-۱}^{k} (-۱)^{k-j} \begin{bmatrix} k-۱ \\ j+۱ \end{bmatrix} = ۰ . \qquad (۱۲–۳)$$

برهان. با جدا کردن دو جملهٔ اول این مجموع و بکارگیری رابطهٔ (۳–۱۰) ، داریم

$$\begin{aligned} (-۱)^{k-\bar{\mu}+۱} \begin{bmatrix} k-۱ \\ \tilde{\mu} \end{bmatrix} + (-۱)^{k-\bar{\mu}} \begin{bmatrix} k-۱ \\ \tilde{\mu}+۱ \end{bmatrix} + \sum_{j=\bar{\mu}+۱}^{k} (-۱)^{k-j} \begin{bmatrix} k-۱ \\ j+۱ \end{bmatrix} \\ = (-۱)^{k-\bar{\mu}} \begin{bmatrix} k-۱ \\ \tilde{\mu}+۱ \end{bmatrix} w_{n_۱+۱} + \sum_{j=\bar{\mu}+۱}^{k} (-۱)^{k-j} \begin{bmatrix} k-۱ \\ j+۱ \end{bmatrix} . \end{aligned}$$

مجدداً با جدا کردن یک جملهٔ دیگر از مجموع باقی‌مانده و انجام محاسبات مشابه داریم

$$\begin{aligned} (-۱)^{k-\bar{\mu}} \begin{bmatrix} k-۱ \\ \tilde{\mu}+۱ \end{bmatrix} w_{n_۱+۱} + (-۱)^{k-\bar{\mu}-۱} \begin{bmatrix} k-۱ \\ \tilde{\mu}+۲ \end{bmatrix} + \sum_{j=\bar{\mu}+۲}^{k} (-۱)^{k-j} \begin{bmatrix} k-۱ \\ j+۱ \end{bmatrix} \\ = (-۱)^{k-\bar{\mu}-۱} \begin{bmatrix} k-۱ \\ \tilde{\mu}+۲ \end{bmatrix} w_{n_۱+۲} + \sum_{j=\bar{\mu}+۲}^{k} (-۱)^{k-j} \begin{bmatrix} k-۱ \\ j+۱ \end{bmatrix} . \end{aligned}$$

به همین ترتیب، این روند را ادامه می‌دهیم تا تمام مجموع صفر شود. مرحله آخر به صورت زیر خواهد بود:

$$(-۱)^{k-k+۱} \begin{bmatrix} k-۱ \\ k \end{bmatrix} w_{n_۲-۱} + \begin{bmatrix} k-۱ \\ k+۱ \end{bmatrix} = -\frac{۱}{w_{n_۲-۱}} w_{n_۲-۱} + ۱ = ۰ .$$

□

لم ۶.۳ اگر دنبالهٔ گره‌ها $\mathcal{U}$ یکنواخت باشد، داریم

$$\begin{bmatrix} k-۱ \\ j+۱ \end{bmatrix} = \binom{k}{j} , \qquad ۰ \le j \le k . \qquad (۱۳–۳)$$

برهان. ابتدا توجه کنید که در دنبالهٔ گره‌های یکنواخت، ضرایب درج گره به سادگی به صورت $w_{j+J} = (k-j)/k$، وقتی $0 \le j \le k$، تعیین می‌شوند. حال بر اساس تعریف ضرایب براکتی داریم:

اگر $0 \le j \le k-2$، آنگاه

$$\begin{bmatrix} k-1 \\ j+1 \end{bmatrix} = \frac{1}{w_{j+J}} \prod_{i=j+J+1}^{n_2-1} \frac{1-w_i}{w_i} = \frac{k}{k-j} \prod_{i=j+1}^{k-1} \frac{i}{k-i} = \binom{k}{j} .$$

اگر $j = k-1$، آنگاه $\begin{bmatrix} k-1 \\ k \end{bmatrix} = \frac{1}{w_{n_2-1}} = k = \binom{k}{k-1}$.

و اگر $j = k$، آنگاه $\begin{bmatrix} k-1 \\ k+1 \end{bmatrix} = \frac{1}{1-w_{n_2}} = 1 = \binom{k}{k}$. □

۳–۲ حذف گره مجازی

همانطور که در بخش ۳–۱–۱ در الگوریتم کاهش درجه در منحنی بزیه مشاهده کردید، چندضلعی کنترلی تظریف شده از دو مسیر، برونیابی شد. در الگوریتم حذف گره که نسخهٔ Bـاسپلاینی الگوریتم کاهش درجه است نیز چندضلعی کنترلی تظریف شده $\{\tilde{c}_i\}_{i=1}^{n+1}$ در رابطهٔ (۲–۲۰) ، را یک بار از ابتدا به انتها و بار دیگر در جهت عکس با ضرایب درج گره w_i برونیابی می‌کنیم. این عمل از دیدگاه ماتریسی با حل دو دستگاه خطی که به ترتیب با صرفنظر کردن از معادلهٔ اول و آخر دستگاه $A\mathbf{c} = \tilde{\mathbf{c}}$ بدست می آیند، انجام می‌شود:

$$c_i^I = \begin{cases} \tilde{c}_i , & 1 \le i \le n_1 , \\ \frac{1}{w_i}\tilde{c}_i + \left(1 - \frac{1}{w_i}\right) c_{i-1}^I , & n_1+1 \le i \le n_2-1 , \\ \tilde{c}_{i+1} , & n_2 \le i \le n . \end{cases} \qquad (۳–۱۴)$$

$$c_i^{II} = \begin{cases} \tilde{c}_{i+1} , & n \ge i \ge n_2-1 , \\ \frac{1}{1-w_{i+1}}\tilde{c}_{i+1} + \left(1 - \frac{1}{1-w_{i+1}}\right) c_{i+1}^{II} , & n_2-2 \ge i \ge n_1 , \\ \tilde{c}_i , & n_1-1 \ge i \ge 1 . \end{cases} \qquad (۳–۱۵)$$

$\{c_i^I\}_{i=1}^n$ و $\{c_i^{II}\}_{i=1}^n$ به عنوان چندضلعیهای حاصل از حذف گره مجازی، شناخته می‌شوند. نکته اساسی در اینجاست که با وجود اینکه $\{c_i^I\}_{i=1}^n$ و $\{c_i^{II}\}_{i=1}^n$ دو جواب خاص مسأله حذف هستند، به وسیلهٔ آنها می‌توان ردهٔ وسیعی از انتخابهای ممکن برای مسأله حذف را در اختیار داشت. به طور دقیقتر، با انتخاب

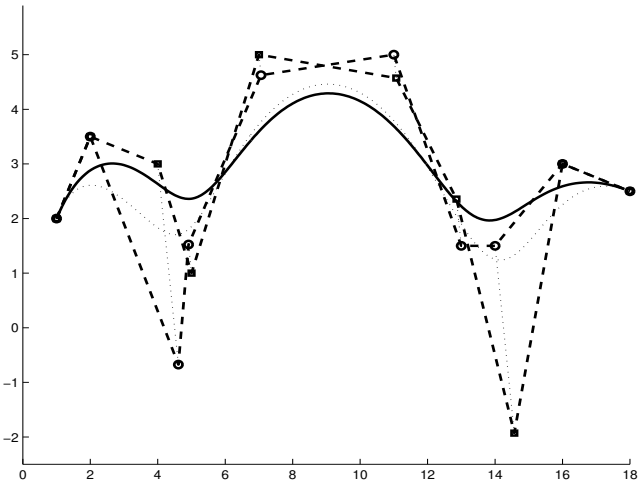
$k - \tilde{\mu} + ۱$ مقدار حقیقی $\lambda_{\tilde{\mu}-۱}, \lambda_{\tilde{\mu}}, \ldots, \lambda_{k-۱}$، یک جواب تقریبی برای چندضلعی اولیه $\{c_i\}_{i=۱}^{n}$، به صورت زیر معرفی می‌شود:

$$\bar{c}_i \;=\; \begin{cases} \tilde{c}_i \;, & ۱ \le i \le n_۱ - ۱ \;, \\ (۱ - \lambda_{i-J})\, c_i^I \;+\; \lambda_{i-J}\, c_i^{II} \;, & n_۱ \le i \le n_۲ - ۱ \;, \\ \tilde{c}_{i+۱} \;, & n_۲ \le i \le n \;. \end{cases} \qquad (۱۶-۳)$$

در اینجا نیز همانطور که ملاحظه می‌کنید، عمدتاً مؤلفه‌های بردار $\bar{\mathbf{c}}$ که به صورت

$$\bar{\mathbf{c}} \;=\; [\bar{c}_{n_۱} \;\cdots\; \bar{c}_{n_۲-۱}]^T \qquad (۱۷-۳)$$

تعریف می‌شود، در حین عمل حذف گره مورد توجه هستند. ابعاد این بردار را با ابعاد $\mathbf{c}$ در (۳–۱) مقایسه کنید.



شکل ۳–۲: نمونه‌ای از حذف گره مجازی: Bـاسپلاین مرتبهٔ ٦ ظریف با (منحنی پررنگ)، رئوس چندضلعی $\mathbf{c}^I$ با (مربعهای کوچک)، رئوس چندضلعی $\mathbf{c}^{II}$ با (دایره‌های کوچک) نشان داده شده‌اند. پاره‌خطهای نقطه‌چین مکان هندسی ضرایب $\bar{c}_i$ را وقتی $۰ \le \lambda_i \le ۱$ نشان می‌دهد. منحنی نقطه‌چین یک منحنی از خانوادهٔ منحنیهای متناظر با ضرایب $\bar{c}_i$ روی این مکان هندسی است.

مثال ۷.۳ در شکل ۳–۲، چندضلعیهای حاصل از حذف مجازی گره $\tilde{u}_۹ = ۷$ از دنباله گره‌های $\{۰, ۰, ۰, ۰, ۰, ۰, ۰, ۳, ۵, ۷, ۸, ۱۲, ۱۵, ۱۵, ۱۵, ۱۵, ۱۵, ۱۵\}$ مربوط به یک منحنی Bـاسپلاین مرتبهٔ ٦ نمایش داده شده‌است. در این شکل با انتخاب $\lambda_۰ = \lambda_۱ = ۰/۵$ و $\lambda_۲ = ۰/۱$ و $\lambda_۳ = ۰/۹$ و $\lambda_۴ = \lambda_۵ = ۰/۵$ یک منحنی به عنوان یک منحنی نوعی حاصل از حذف گره رسم شده‌است. △

در این وضعیت اگر قرار دهیم

$$\tilde{Q}(t) = \sum_{i=1}^{n+1} \tilde{c}_i \, \tilde{B}_i^k(t) \quad , \quad \bar{Q}(t) = \sum_{i=1}^{n} \bar{c}_i \, B_i^k(t) \qquad (۱۸-۳)$$

با توجه به اینکه تنها تعداد کمی از رئوس کنترلی در $\{\bar{c}_i\}_{i=1}^{n}$، متمایز از $\{\tilde{c}_i\}_{i=1}^{n+1}$ هستند و به واسطهٔ ویژگی کنترلپذیری موضعی منحنیهای Bـاسپلاین، به طور بدیهی می‌توان نشان داد که دو منحنی $\tilde{Q}(t)$ و $\bar{Q}(t)$ بجز روی بازهٔ $\mathcal{H} = (\tilde{u}_{n_1} \ \tilde{u}_{n_2+k})$، در سایر دامنه بر هم منطبق هستند. امّا روی $\mathcal{H}$ عموماً این دو منحنی متمایز هستند. لم زیر به طور بدیهی نحوهٔ کوچکتر نمودن $\mathcal{H}$ را بیان می‌کند.

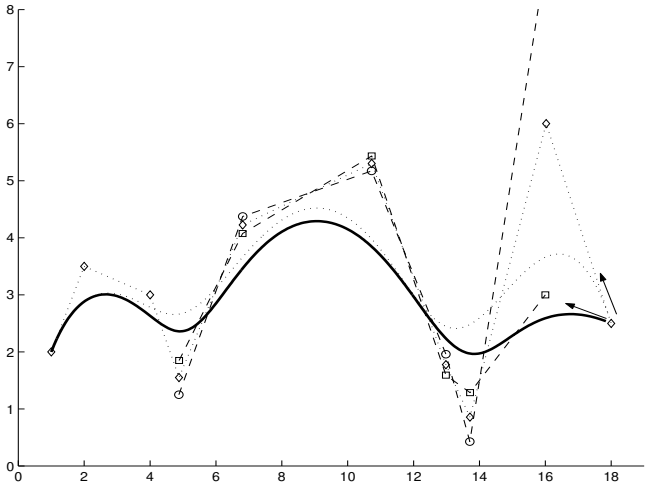
لم ۸.۳ اگر برای $k - m_2 \le j \le k - 1$ و برای $\lambda_j = 0$، قرار دهیم $\tilde{\mu} - 1 \le j \le \tilde{\mu} - 2 + m_1$ قرار دهیم $\lambda_j = 1$ آنگاه دو منحنی $\tilde{Q}(t)$ و $\bar{Q}(t)$ بجز روی $\mathcal{H}_{m_1,m_2} = (\tilde{u}_{n_1+m_1} \ \tilde{u}_{n_2+k-m_2})$، روی سایر دامنه بر هم منطبق هستند.

بدین ترتیب می‌توان به وسیلهٔ انتخاب مناسب مقادیر m_1, m_2 ترتیبی داد تا دو منحنی $\bar{Q}$ و $\tilde{Q}$ و مشتقاتشان تا حداکثر مرتبهٔ ممکن یعنی $k-2$، در لبه‌های دامنه $\mathcal{I} = [\tilde{u}_k \ \tilde{u}_{n+2})$ با هم برابر باشند. این خواسته از آنجا ناشی می‌شود که در بسیاری موارد نمی‌خواهیم دو سر منحنی قبل و بعد از حذف گره تغییر مکان و یا تغییر شکل دهد (به شکلهای ۳-۳، ۳-۴ و ۳-۵ نگاه کنید). به وضوح این خواسته در مواردی که گره حذف شده دور از دو سر دامنه باشد، خودبخود ارضاء می‌شود.

به طور کلی روش زیر برای انتخاب مناسب m_1, m_2 پیشنهاد می‌شود. اول آنکه خود را فقط به دنباله گره‌هایی که در آنها گره‌های اول و آخر k بار تکرار شده‌اند، مقید می‌کنیم. به علاوه با انتخاب عدد β، که $-1 \le \beta \le 1 + \lceil \frac{1}{2} k \rceil$، به عنوان مرتبهٔ مشتقات برابر قرار داده شده در دو لبهٔ $\mathcal{I}$، قرار می‌دهیم

$$m_1 = \max(0, k - \tilde{\delta} + \beta) \quad , \quad m_2 = \max(0, \tilde{\delta} - \tilde{\mu} - n + 1 + \beta) \, . \qquad (۱۹-۳)$$

با این نحوهٔ انتخاب ملاحظه می‌شود که تنها وقتی $\tilde{\delta} < k + \beta$ یا $\tilde{\delta} > n + \tilde{\mu} - 1 + \beta$ ، m_1 یا m_2 بیشتر از صفر می‌شوند، و وقتی گره به قدر کافی از دو سر دامنه دور باشد، داریم $m_1 = 0, m_2 = 0$.



شکل ۳—۳: عدم انطباق مشتقات اول و دوم منحنی ظریف و درشت در انتهای دامنه به خاطر انتخاب نامناسب $m_1 = m_2 = 0$: منحنی ظریف کلفت‌تر و منحنی درشت نازک‌تر رسم شده‌اند. چندضلعیهای خط‌چین و نقطه‌چین از حذف مجازی گره $\tilde{u}_{11} = 12$ بدست آمده‌اند و تمام ضرایب λ_i برابر ۰/۵ انتخاب شده‌اند.

۳—۲—۱ ضریب خطای حذف گره

تعریف ۹.۳ به ازای هر $\tilde{\mathbf{c}}$ ، $\tilde{U}$ و $\tilde{\delta}$، عدد D را که به صورت

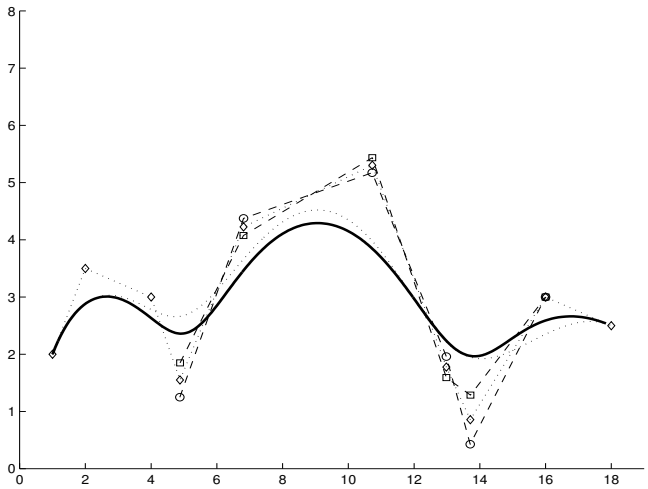
$$D = \sum_{j=\bar{\mu}-1}^{k} (-1)^{k-j} \begin{bmatrix} k-1 \\ j+1 \end{bmatrix} \tilde{c}_{j+J} \qquad (۲۰—۳)$$

تعریف می‌شود، ضریب خطای حذف گره $\tilde{u}_{\bar{\delta}}$ از دنباله گره‌های $\tilde{U}$ برای $\tilde{\mathbf{c}}$ می‌نامیم.

عدد D نقش بسیار مهمی در بیان تابع خطا و مطالب مربوط به تعیین بهترین تقریب ایفاء می‌کند. در ادامه نشان داده می‌شود که D، به نوعی، نمایانگر توانایی ذاتی $\tilde{\mathbf{c}}$، برای پایا ماندن بعد از عمل حذف گره است.

لم ۱۰.۳ فرض کنید $\bar{\mathbf{c}}$ و $\tilde{\mathbf{c}}$ با رابطهٔ (۳—۱۶) به هم مربوط شده‌اند، آنگاه

$$\tilde{\mathbf{c}} - A\,\bar{\mathbf{c}} = (\mathbf{e} - A\,\Lambda)\,D \qquad (۲۱—۳)$$



شکل ۳–۴: انطباق مشتق اول منحنی ظریف و درشت با انتخاب $m_۲ = ۱$: هنوز انحنای دو منحنی در انتهای دامنه متمایز از یکدیگر است.

که $\mathbf{e} = [۰ \cdots ۰ \ ۱]^T$ و $\Lambda = [\widetilde{\lambda}_{\bar{\mu}-۱} \cdots \widetilde{\lambda}_{k-۱}]^T$ و

$$\widetilde{\lambda}_j \ = \ (-۱)^{k-j+۱} \ \frac{۱}{w_{j+J}} \ \begin{bmatrix} k-۱ \\ j+۱ \end{bmatrix}^{-۱} \ \lambda_j \ , \quad \tilde{\mu}-۱ \le j \le k-۱ \ . \qquad (۲۲–۳)$$

برهان. در ابتدا دو رابطهٔ صریح برحسب $\tilde{\mathbf{c}}$ برای $\mathbf{c}^I, \mathbf{c}^{II}$ بدست می‌آوریم، که به سادگی به وسیلهٔ استقراء قابل اثبات است:

$$c^I_{n_۲-j} = (-۱)^j \ \frac{۱}{w_{n_۲-j}} \ \begin{bmatrix} k-۱ \\ k-j+۱ \end{bmatrix}^{-۱} \ \sum_{s=\bar{\mu}-۱}^{k-j} (-۱)^{k-s} \ \begin{bmatrix} k-۱ \\ s+۱ \end{bmatrix} \ \tilde{c}_{s+J} \qquad (۲۳–۳)$$

$$c^{II}_{n_۲-j} = -(-۱)^j \ \frac{۱}{w_{n_۲-j}} \ \begin{bmatrix} k-۱ \\ k-j+۱ \end{bmatrix}^{-۱} \ \sum_{s=k-j+۱}^{k} (-۱)^{k-s} \ \begin{bmatrix} k-۱ \\ s+۱ \end{bmatrix} \ \tilde{c}_{s+J} \ . \qquad (۲۴–۳)$$

در اینجا $۱ \le j \le k - \tilde{\mu} + ۱$. سپس حالت میانی در رابطهٔ (۳–۱۶) را به صورت

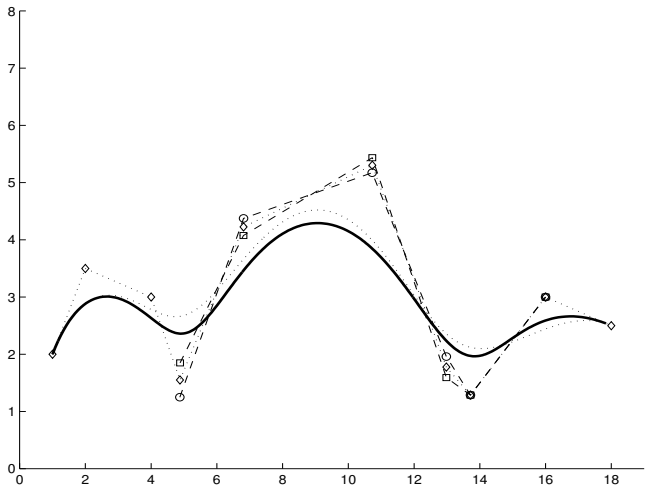
$$\bar{c}_{n_۲-j} = c^I_{n_۲-j} + \lambda_{k-j}(c^{II}_{n_۲-j} - c^I_{n_۲-j}) \ , \qquad ۱ \le j \le k - \tilde{\mu} + ۱$$

بازنویسی می‌کنیم. با ترکیب رابطهٔ اخیر با (۳–۲۳) و (۳–۲۴) ، داریم

$$\bar{c}_{n_۲-j} = c^I_{n_۲-j} - \lambda_{k-j} \frac{(-۱)^{j+۱}}{w_{n_۲-j}} \ \begin{bmatrix} k-۱ \\ k-j+۱ \end{bmatrix}^{-۱} \ \sum_{s=\bar{\mu}-۱}^{k} (-۱)^{k-j} \ \begin{bmatrix} k-۱ \\ s+۱ \end{bmatrix} \ \tilde{c}_{s+J}$$

که با نمادگذاری داده شده در (۳–۲۲) و (۳–۲۰) ، رابطهٔ اخیر به صورت

$$\bar{c}_{n_۲-j} = c^I_{n_۲-j} + \widetilde{\lambda}_{k-j} \, D \ , \qquad ۱ \le j \le k - \tilde{\mu} + ۱$$



شکل ۳–۵: انطباق مشتقات اول و دوم منحنی ظریف و درشت با انتخاب مناسب $m_2 = 2$: با این انتخاب داریم $\lambda_4 = \lambda_5 = 1$.

نوشته می‌شود و آن را می‌توان در فرم برداری به صورت

$$\bar{\mathbf{c}} = \mathbf{c}^I + \Lambda\, D \qquad (۲۵–۳)$$

خلاصه کرد.

اگر کمی دقیقتر به مؤلفه‌های برادر $\tilde{\mathbf{c}} - A\mathbf{c}^I$ نگاه کنید، متوجه می‌شوید که جز مؤلفهٔ آخر مابقی مؤلفه‌ها صفر هستند. به طور دقیق، از رابطهٔ (۳–۱۴) داریم

$$\begin{aligned} \tilde{\mathbf{c}} - A\mathbf{c}^I &= \mathbf{e}(\tilde{c}_{n_2} - c^I_{n_2-1}) \\ &= \mathbf{e}(c^{II}_{n_2-1} - c^I_{n_2-1}) \\ &= \mathbf{e}\, D \qquad (۲۶–۳) \end{aligned}$$

که آخرین عبارت با استفاده از (۳–۱۴) و (۳–۱۵) به ازای $j = 1$ بدست می‌آید. با ترکیب دو رابطهٔ (۳–۲۵) و (۳–۲۶) حکم (۳–۲۱) بدست می‌آید. □

تبصره. کلیهٔ روابط فوق برای وقتی که مؤلفه‌های $\tilde{\mathbf{c}}$ و $\bar{\mathbf{c}}$، خود بردارهایی در $\mathbb{R}^S$ هستند نیز معتبر است. در بخش ۴–۲ به طور اساسی از بردار D در الگوریتمهای تجزیه و ترکیب استفاده خواهد شد.

در قضیهٔ بعدی، این تعبیر شهودی که هرقدر اندازهٔ D کوچکتر باشد، خطای بین دو منحنی قبل و بعد از حذف گره کمتر است، برای یک حالت تعیین کننده ثابت می‌شود.

قضیه‌ی ۱۱.۳ گره $\tilde{u}_{\bar{\delta}}$ به طور دقیق وقتی و تنها وقتی از منحنی $\tilde{Q}$ قابل حذف است که $D = 0$.

برهان. به وضوح، حذف دقیق گره $\tilde{u}_{\bar{\delta}}$ از منحنی $\tilde{Q}$ معادلست با اینکه $\tilde{\mathbf{c}} = A\bar{\mathbf{c}}$. بنابراین اگر $D = 0$، بر اساس رابطهٔ (۳–۲۱) مشاهده می‌شود که $\tilde{\mathbf{c}} = A\bar{\mathbf{c}}$ و به اصطلاح گره $\tilde{u}_{\bar{\delta}}$ به طور مصنوعی در منحنی، درج شده است.

از طرف دیگر، وقتی $\tilde{\mathbf{c}} - A\bar{\mathbf{c}} = 0$، براساس لم ۲.۳ می‌بینیم که دستگاه $\mathbf{e} = A\Lambda$ برای هیچ انتخاب Λ دارای جواب نیست و در نتیجه $D = 0$. □

بر اساس همین ویژگی و در مواردی که اندازهٔ $\|D\|_2$ کمتر از دقت کامپیوتر در محاسبات اعشاری است، می‌توان یک الگوریتم سادهٔ حذف گره بر اساس (۳–۱۴) یا (۳–۱۵) و یا با انتخاب هر مقدار دلخواه برای ضرایب وزنی λ_i، با استفاده از رابطهٔ (۳–۱۶) پیاده‌سازی کرد. توجه کنید که در این وضعیت $\mathbf{c}^I \simeq \mathbf{c}^{II} \simeq \bar{\mathbf{c}} \simeq \mathbf{c}$. الگوریتمهای سریعی برای حذف تمام گره‌های تقریباً مصنوعی یک منحنی یا رویهٔ Bـاسپلاین توسط **تیلر** بر اساس همین ایده در [۳۹] ارائه شده‌اند.

در این بخش دو انتخاب خاص $\{c_i^I\}_{i=1}^n$ و $\{c_i^{II}\}_{i=1}^n$ به عنوان چندضلعی کنترلی منحنی حاصل از حذف گره، ارائه شد امّا هنوز این مسأله که چه انتخابی از λ_i ها ما را به بهترین منحنی می‌رساند، بی‌پاسخ ماند.

۳–۳ حذف گره بهترین تقریب

آنچه در این بخش به آن پرداخته می‌شود، تکمیل الگوریتم حذف گره، با بدست آوردن ضرایب وزنی λ_i از طریق مینیمم سازی مقدار برخی توابع خطا و بکارگیری آنها در رابطهٔ(۳–۱۶) برای بدست آوردن یک

چندضلعی کنترلی $\{\bar{c}_i\}_{i=1}^{n}$ به عنوان نتیجهٔ الگوریتم حذف گره است.

۳–۳–۱ تعریف یک تابع خطا

منظور از تابع خطا، تابعی است که میزان تغییر منحنی را بعد از حذف گره نشان می‌دهد. برای تعریف این تابع ابتدا باید یک متر مناسب انتخاب کنیم. از دیدگاه هندسی، متر هاوسدورف یک انتخاب مناسب برای سنجش اختلاف دو منحنی به نظر می‌رسد. اخیراً نیز دِگِن [۱۷] یک متر موسوم به متر نرمال معرفی کرده است که آن نیز معمولاً برای دو منحنی هموار، با متر هاوسدروف معادل است، هرچند اینگونه نحوهٔ تعریف منجر به حل عددی مسائل غیرخطی می‌شود.

در عوض، در اینجا یک متر بسیار ساده‌تر برای اندازه‌گیری اختلاف دو منحنی به صورت زیر به کار برده می‌شود:

$$\mathrm{dist}(\tilde{Q},\bar{Q}) \;=\; \max\{\|\tilde{Q}(t)-\bar{Q}(t)\|_2 \;:\; t\in\mathcal{I}=[\tilde{u}_k\;\tilde{u}_{n+2})\}$$

که حداکثر فاصلهٔ اقلیدسی بین نقاط هم پارامتر دو منحنی است. بنابراین در این متر دو منحنی هم شکل از دید هندسی، لزوماً دارای فاصلهٔ صفر نیستند. با این حال جواب مسألهٔ تقریب یک منحنی در این متر، قویتر از حالت هندسی است و تقریباً آنرا نیز در بر می‌گیرد.

بر اساس رابطهٔ (۳–۱۶) بین $\tilde{\mathbf{c}}$ و $\bar{\mathbf{c}}$ و لم ۸.۳، خواهیم داشت

$$\mathrm{dist}(\tilde{Q},\bar{Q}) \;=\; \max\{\|\tilde{\mathbf{B}}^T(t)E(\tilde{\mathbf{c}}-A\bar{\mathbf{c}})\|_2 \;:\; t\in\mathcal{H}_{m_1,m_2}=[\tilde{u}_{n_1+m_1}\;\tilde{u}_{n_2+k-m_2})\} \qquad (۳–۲۷)$$

که در اینجا ماتریس قطری E با ابعاد $k-\tilde{\mu}+2$ به صورت

$$E=\mathrm{diag}(\underbrace{0,0,\ldots,0}_{m_1},\;\underbrace{1,1,\ldots,1}_{M+1},\;\underbrace{0,\ldots,0,0}_{m_2}) \qquad (۳–۲۸)$$

داده می‌شوند. به علاوه، این قید که m_1 معادلهٔ اول و همچنین m_2 معادلهٔ آخر در دستگاه ناسازگار $\tilde{\mathbf{c}}=A\bar{\mathbf{c}}$، در هر صورت، باید برقرار باشند:

$$\bar{E}.(\tilde{\mathbf{c}}-A\bar{\mathbf{c}})=0 \qquad \text{که } \bar{E}=I_{k-\tilde{\mu}+1}-E\,. \qquad (۳–۲۹)$$

در صورت استفاده از رابطهٔ (۳–۲۱) ، تابع خطای فوق به طور ساده‌تری فرمولبندی می‌شود:

$$\mathrm{dist}(\tilde{Q},\bar{Q}) \;=\; R\,\|D\|_{۲} \qquad \text{که } R=\max\{|\tilde{\mathbf{B}}^{T}(t)E(\mathbf{e}-A\Lambda)| \;:\; t\in\mathcal{H}_{m_{۱},m_{۲}}\}\,. \qquad (۳۰–۳)$$

نکتهٔ جالب توجه در این طرح آنست که مسألهٔ مینیمم‌سازی با متغیرهای برداری به شکل یک مسألهٔ مینیمم‌سازی با متغیرهای حقیقی تقلیل می‌یابد:

$$\min_{\bar{c}_i\in\mathbb{R}^{S}}\mathrm{dist}(\tilde{Q},\bar{Q})=\min_{\lambda_i\in\mathbb{R}}R\,.\,\|D\|_{۲}\,. \qquad (۳۱–۳)$$

آنچنانکه در روشهای کلاسیک مسائل بهترین تقریب متداول است، ابتدا سعی می‌کنیم، به جای جستجوی مستقیم یک جواب، با یافتن برخی کرانها برای تابع خطا، آن را در زیر این کرانها محدود کنیم. در اینجا دو کران بالا به ترتیب بر اساس نرمهای برداری L_{∞} و $L_{۲}$ برای R می‌توان در نظر گرفت:

$$R\le\|E(\mathbf{e}-A\Lambda)\|_{\infty} \qquad (۳۲–۳)$$

$$R\le\|E(\mathbf{e}-A\Lambda)\|_{۲} \qquad (۳۳–۳)$$

کران اول مستقیماً از ویژگی پوشش محدب در منحنیهای Bـاسپلاین نتیجه می‌شود، و کران دوم توسط شَبَک [۳۵] با بررسی خطا به وسیلهٔ چندضلعی‌های کنترلی بدست آمده است.

با یافتن مینیمم مقدار این دو کران به ازای بردار Λ (ضرایب وزنی λ_i به نوعی در این بردار وجود دارند، در رابطهٔ (۳–۲۲))، می‌توان انتظار داشت که مقدار خطای بین منحنی اولیه و منحنی بعد از حذف گره تا اندازهٔ نسبتاً قابل قبولی به سمت مقدار مینیمم دقیق، کاهش یافته است. بنابراین دو روشی که عمل حذف گره را در جهت مینیمم نمودن این دو کران انجام می‌دهند، به عنوان دو روش حذف گره که در حد خود مفید هستند، به کار خواهیم برد. عموماً مسائل مینیمم سازی این دو نرم برداری به ترتیب بعنوان تقریب L_{∞} گسسته و تقریب $L_{۲}$ گسسته در مباحث نظریهٔ تقریب شناخته می‌شوند. از طرف دیگر مسألهٔ اصلی یافتن مقدار مینیمم برای تابع R به عنوان تقریب L_{∞} پیوسته شناخته می‌شود. هر سه مسألهٔ مینیمم سازی به طور مقید، با قید (۳–۲۹) روی متغیرهای λ_i انجام می‌شود.

قبل از پرداختن به هر یک از روشهای مزبور، قضیهٔ اساسی زیر را اثبات می‌کنیم.

قضیه‌ی ۱۲.۳ فرض کنید $\mathbf{h} = [h_{\tilde{\mu}-1} \cdots h_k]^T$ یک بردار دلخواه داده شده‌است. آنگاه دستگاه

$$\mathbf{e} - A\Lambda = \alpha \mathbf{h} \,, \quad \alpha \in I\!R \tag{۳۴–۳}$$

شامل $k - \tilde{\mu} + 2$ معادلهٔ خطی وابسته به $k - \tilde{\mu} + 2$ مجهول $\{\lambda_i\}_{i=\tilde{\mu}-1}^{k-1}$ و α، به طور یکتا قابل حل است اگر و فقط اگر

$$\gamma = \sum_{j=\tilde{\mu}-1}^{k} (-1)^{k-j} \begin{bmatrix} k-1 \\ j+1 \end{bmatrix} h_j \neq 0 \tag{۳۵–۳}$$

و در اینصورت جواب به طور مشخص توسط رابطهٔ زیر داده می‌شود

$$\lambda_i = \gamma^{-1} \sum_{j=\tilde{\mu}-1}^{i} (-1)^{k-j} \begin{bmatrix} k-1 \\ j+1 \end{bmatrix} h_j \,, \qquad \tilde{\mu} - 1 \leq i \leq k-1 \,, \tag{۳۶–۳}$$

$$\alpha = \gamma^{-1} \,. \tag{۳۷–۳}$$

برهان. ابتدا، ماتریس افزودهٔ $\widetilde{A} = [A \; h]$ را با محاسبهٔ دترمینان آن مورد بررسی قرار می‌دهیم:

$$\begin{aligned} \det \widetilde{A} &= \sum_{j=\tilde{\mu}-1}^{k} (-1)^{k-j} h_j \det A_{0,0:j+1} \\ &= \prod_{i=\tilde{\mu}}^{k-1} w_{i+J} \sum_{j=\tilde{\mu}-1}^{k} (-1)^{k-j} \begin{bmatrix} k-1 \\ j+1 \end{bmatrix} h_j \\ &= \gamma \prod_{i=\tilde{\mu}}^{k-1} w_{i+J} \end{aligned}$$

که در اینجا از رابطهٔ (۳–۸) در لم ۲.۳ استفاده شد. بدین ترتیب ملاحظه می‌شود که دستگاه مذکور قابل حل است، اگر و تنها اگر $\gamma \neq 0$.

حال برای نشان دادن اینکه جواب پیشنهاد شده توسط (۳–۳۶) و (۳–۳۷) دستگاه را صدق می‌دهد، قرار می‌دهیم $\alpha = \gamma^{-1}$ و بدین ترتیب دستگاه (۳–۳۴) به صورت $k - \tilde{\mu} + 2$ معادلهٔ زیر در می‌آید

$$\begin{aligned} -\widetilde{\lambda}_{\tilde{\mu}-1} &= \gamma^{-1} h_{\tilde{\mu}-1} \,, \\ -(1 - w_{i+J})\widetilde{\lambda}_{i-1} - w_{i+J}\widetilde{\lambda}_i &= \gamma^{-1} h_i \,, \qquad \tilde{\mu} \leq i \leq k-1 \,, \\ 1 - \widetilde{\lambda}_{k-1} &= \gamma^{-1} h_k \,. \end{aligned}$$

با بکارگیری (۳–۲۲) و جایگذاری روابط بدست آمده در (۳–۳۵) و (۳–۳٦) ، با یک جایگذاری پیشرو روی این دستگاه، صحت حکم ثابت می‌شود. □

از این قضیهٔ اساسی مکرراً در هر یک از بخشهای بعدی استفاده می‌کنیم. بدین ترتیب که هر یک از مسائل تقریب را به صورت چنین دستگاهی فرمولبندی می‌کنیم و پس از تعیین $\mathbf{h}$ بر اساس نوع مسأله، ضرایب وزنی $\{\lambda_i\}_{i=\bar{\mu}-1}^{k}$ را از طریق قضیهٔ فوق بدست می‌آوریم.

در مورد اینکه آیا این ضرایب وزنی، هر نقطهٔ $\bar{c}_i$ را روی پاره‌خط $c_i^I c_i^{II}$ درونیابی می‌کنند یا نه، تا اینجا به طور دقیق مطلبی گفته نشده است. هرچند بر اساس قضیهٔ پیش، و به ازای تقریباً هر نوع انتخاب برای $\mathbf{h}$، این ویژگی دور از انتظار نیست.

نتیجه‌ی ۱۳.۳ اگر مؤلفه‌های بردار $\mathbf{h}$ متناوباً تغییر علامت دهند و رابطهٔ (۳–۳۵) را صدق دهند، آنگاه برای جواب بدست آمده در (۳–۳٦) داریم

$$0 \leq \lambda_{\bar{\mu}-1} \leq \lambda_{\bar{\mu}} \leq \cdots \leq \lambda_{k-2} \leq \lambda_{k-1} \leq 1 \ . \qquad (۳–۳۸)$$

اگر $\mathbf{h}$ به طوری که شرایط قضیهٔ ۱۲.۳ را صدق دهد، برای یک روش حذف گره در اختیار باشد، با استفاده از رابطهٔ (۳–۲۱) و جایگذاری طرف راست آن با (۳–۳۴) ، یک رابطهٔ مفید برای بازسازی منحنی ظریف از روی منحنی درشت، تنها با در دست داشتن ضریب خطای حذف گره D بدست می‌آید:

$$\tilde{\mathbf{c}} \ = \ A\bar{\mathbf{c}} + \alpha \mathbf{h} D \ . \qquad (۳–۳۹)$$

این رابطه را با توجه به (۳–۱٦) ، با (۳–۴) مقایسه کنید. از رابطهٔ (۳–۳۹) به طور اساسی در بخش ۴–۲، در الگوریتم بازسازی استفاده خواهد شد.

۳–۳–۲ تقریب L_∞ گسسته

حل دستگاه ناسازگار $\mathbf{e} - A\Lambda = 0$ تحت قیود خطی $\bar{E}(\mathbf{e} - A\Lambda) = 0$، معادلست با حل یک دستگاه ناسازگار کوچکتر بدون منظور کردن هیچ قیدی، که این دستگاه کوچکتر از افزودن قیود خطی مزبور به دستگاه اول بدست می آید. با انجام این کار، ماتریس درج گره کاهش‌یافته $A_{m_1,m_2} = EA$ به عنوان ماتریس مورد نظر بدست می آید.

بلافاصله مشاهده می‌شود که L_∞ـجواب این دستگاه (کوچک) ناسازگار (در برخی موارد جواب چبیشف خوانده می‌شود) به طور یکتا بدست می آید. این امر به این دلیل است که ماتریس A_{m_1,m_2} در شرط هار[۶]، مبنی براینکه تمام کهادهای مرتبهٔ M، غیر صفر باشند، صدق می‌کند (طبق لم ۲.۳).

به علاوه به ازای L_∞ـجواب، مقدار خطای $\widehat{\alpha}_\infty = \|E(\mathbf{e} - A\Lambda)\|_\infty$ به طور تناوبی روی هر یک از معادلات این دستگاه خطی ناسازگار اتفاق می‌افتد. بنابراین مسألهٔ یافتن L_∞ـجواب برای دستگاه $E(\mathbf{e} - A\Lambda)$ معادل می‌شود با حل دستگاه

$$\mathbf{e} - A\Lambda = \widehat{\alpha}_\infty \widehat{\mathbf{h}}_\infty \qquad (۳–۴۰)$$

برای مجهولات $\widehat{\alpha}_\infty$ و Λ، که در آن $\widehat{\mathbf{h}}_\infty = [\widehat{h}_{\tilde{\mu}-1} \cdots \widehat{h}_k]^T$ بردار علامتهاست و به صورت

$$\widehat{h}_i = \begin{cases} (-1)^{k-i}.\,\mathrm{sign}(\det A_{m_1,m_2;i}) = (-1)^{k-i}\,, & \tilde{\mu} - 1 + m_1 \le i \le k - m_2 \\ 0\,, & \text{در غیر اینصورت} \end{cases} \qquad (۳–۴۱)$$

داده می‌شود. اعتبار این انتخاب برای علامت کلی بردار $\widehat{\mathbf{h}}_\infty$ توسط چنی در [۵] به کمک قاعدهٔ کرامر، ثابت شده است.

توجه کنید که با این نحوهٔ تعریف برای $\widehat{\mathbf{h}}_\infty$، قیود $\bar{E}(\mathbf{e} - A\Lambda) = 0$ خودبخود در دستگاه (۳–۴۰) منظور شده‌اند. بدین ترتیب با بکارگیری قضیهٔ ۱۲.۳ برای دستگاه (۳–۴۰) و $\mathbf{h} = \widehat{\mathbf{h}}_\infty$، مسألهٔ حذف گره با تقریب L_∞ گسسته حل می‌شود. ضرایب وزنی λ_i و کران بالای تابع خطا در این روش، به طور مشخص در قضیهٔ زیر بیان می‌شوند.

---

[۶] Haar condition

قضیه‌ی ۱۴.۳ اگر و تنها اگر ضرایب وزنی λ_i با رابطهٔ

$$\lambda_i = \begin{cases} ۰ , & \tilde{\mu} - ۱ \le i \le \tilde{\mu} - ۲ + m_۱ , \\ \widehat{\gamma}_\infty^{-۱} \sum_{j=\bar{\mu}-۱}^{i} \begin{bmatrix} k-۱ \\ j+۱ \end{bmatrix} , & \tilde{\mu} - ۱ + m_۱ \le i \le k - ۱ - m_۲ , \\ ۱ , & k - m_۲ \le i \le k - ۱ \end{cases} \qquad (۴۲-۳)$$

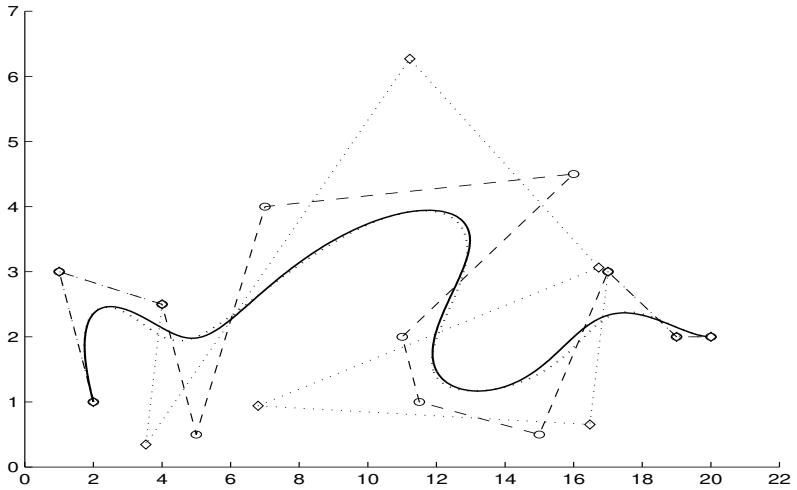
تعیین شوند، که در آن

$$\widehat{\gamma}_\infty = \sum_{j=\bar{\mu}-۱+m_۱}^{k-m_۲} \begin{bmatrix} k-۱ \\ j+۱ \end{bmatrix} \qquad (۴۳-۳)$$

آنگاه برای چندضلعی کنترلی داده شدهٔ $\{\tilde{c}_i\}_{i=۱}^{n+۱}$ با $D \neq ۰$، چندضلعی کنترلی $\{\bar{c}_i\}_{i=۱}^{n}$ که طبق (۳-۱۶) بدست می آید، جواب مینیمم‌کنندهٔ $\|\tilde{\mathbf{c}} - A\bar{\mathbf{c}}\|_\infty$ با قید $\tilde{Q} \equiv \bar{Q}$ روی $\mathcal{I} \setminus \mathcal{H}_{m_۱,m_۲}$ است. به علاوه، داریم

$$\operatorname{dist}(\tilde{Q}, \bar{Q}) \le \|\tilde{\mathbf{c}} - A\bar{\mathbf{c}}\|_\infty = \widehat{\gamma}_\infty^{-۱} \, \|D\|_۲ \, . \qquad (۴۴-۳)$$

تبصره. توجه کنید که برقراری شرط هار و بدست آوردن جواب برای تقریب L_∞ گسسته، تنها برای ماتریس کوچک درج گره $A_{m_۱,m_۲}$ ممکن است و چون A_L در این شرط صدق نمی‌کند، بنابراین لزوماً نمی‌توان یک L_∞ـجواب یکتا برای دستگاه $\tilde{\mathbf{c}}_L = A_L \bar{\mathbf{c}}_L$ بدست آورد.



شکل ۳-۶: حذف گره با تقریب L_∞ گسسته: یک منحنی Bـاسپلاین مرتبه ۶ (منحنی کلفت) و منحنی حاصل از حذف گره با تقریب L_∞ گسسته (منحنی نازک).

مثال ۱۵.۳ در شکل ۳-۶، در یک منحنی Bـاسپلاین مرتبهٔ ۶، گره $\tilde{u}_{۱۰} = ۹$ از دنباله گره‌های $\tilde{U} = \{۰, ۰, ۰, ۰, ۰, ۰, ۳, ۵, ۹, ۹, ۱۴, ۱۵, ۱۷, ۱۷, ۱۷, ۱۷, ۱۷, ۱۷\}$ با روش تقریب L_∞ گسسته

حذف شده‌است. کران بدست آمده برای خطای بین دو منحنی ظریف و درشت در این روش برابر $\widehat{\gamma}_{\infty}^{-1}\|D\|_2 = 1/4820$ است. $\triangle$

۳–۳–۳ تقریب L_2 گسسته

ابتدا لم ضروری زیر را مورد توجه قرار می‌دهیم.

لم ۱۶.۳ فرض کنید $\widetilde{\mathbf{h}}_2 = [\widehat{h}_{\tilde{\mu}-1} \cdots \widehat{h}_k]^T$ یک بردار با مؤلفه‌های $\widehat{h}_j = (-1)^{k-j}\begin{bmatrix} k-1 \\ j+1 \end{bmatrix}$ برای $k \leq j \leq \tilde{\mu}-1$ باشد. آنگاه ماتریس درج گره A در رابطهٔ مشخصهٔ

$$A^T\widetilde{\mathbf{h}}_2 = 0 \qquad (۴۵–۳)$$

صدق می‌کند.

برهان. با ارزیابی سطر به سطر رابطهٔ (۳–۴۵) برای $\tilde{\mu} \leq i \leq k$ و به کمک قاعدهٔ ضرایب براکتی (۳–۱۰) بدست می‌آوریم:

$$\begin{aligned} w_{i+J-1}\widehat{h}_{i-1} + (1-w_{i+J})\widehat{h}_i &= w_{i+J-1}(-1)^{k-i+1}\begin{bmatrix} k-1 \\ i \end{bmatrix} + (1-w_{i+J})(-1)^{k-i}\begin{bmatrix} k-1 \\ i+1 \end{bmatrix} \\ &= \left(w_{i+J-1}(-1)^{k-i+1}\frac{1-w_{i+J}}{w_{i+J-1}} + (1-w_{i+J})(-1)^{k-i}\right)\begin{bmatrix} k-1 \\ i+1 \end{bmatrix} \\ &= 0\,. \end{aligned}$$

□

حال به مسألهٔ مورد بحث یعنی یافتن مینیمم $\|E(\mathbf{e}-A\Lambda)\|_2$ با قید $\bar{E}(\mathbf{e}-A\Lambda)=0$ می‌پردازیم. این مسأله به عنوان مسألهٔ کمترین مربعات خطی با قیود خطی شناخته می‌شود و در مباحث کلاسیک با روشهای مختلف قابل حل است. در اینجا، یک روش مبتنی بر نوع خاصی از معادلات نرمال، برای حل این مسأله ارائه می‌شود.

یکتا بودن جواب در روش زیر به واسطهٔ اینکه A یک ماتریس رتبه کامل است، تضمین می‌شود. برای رهایی از قیود خطی مورد بحث، آنها را به وسیلهٔ یک ضریب وزنی با دستگاه مسألهٔ کمترین مربعات ترکیب

می‌کنیم. این ضرایب در مباحث بهینه‌سازی به عنوان ضرایب جریمه‌ای شناخته شده هستند. برای این کار، E_ε را به عنوان یک ماتریس قطری با ابعاد $k-\tilde{\mu}+۲$ به صورت زیر در نظر می‌گیریم:

$$E_\varepsilon = \mathrm{diag}(\underbrace{\varepsilon,\varepsilon,\ldots,\varepsilon}_{m_۱},\underbrace{۱,۱,\ldots,۱}_{M+۱},\underbrace{\varepsilon,\ldots,\varepsilon,\varepsilon}_{m_۲}) \ . \qquad (۴۶-۳)$$

برای ε ثابت نگهداشته شده، دستگاه خطی وزندار $E_\varepsilon A\Lambda = E_\varepsilon \mathbf{e}$ را در نظر می‌گیریم. دستگاه معادلات نرمال متناظر با این دستگاه به صورت

$$A^T E_\varepsilon^T E_\varepsilon (A\Lambda_\varepsilon - \mathbf{e}) = ۰ \qquad (۴۷-۳)$$

داده می‌شود، که در اینجا Λ_ε نشاندهندهٔ $L_۲$ـجواب این مسأله است. آنچنانکه در [۲۴] اثبات شده است، جواب مسألهٔ مقید $\widehat{\Lambda}$ با حدگیری از $L_۲$ـجواب دستگاه (۳-۴۷) روی ε بدست می‌آید:

$$\widehat{\Lambda} = \lim_{\varepsilon\to\infty} \Lambda_\varepsilon \ . \qquad (۴۸-۳)$$

حال با قرار دادن

$$\widehat{\mathbf{h}}_{۲,\varepsilon} = (E_\varepsilon^T E_\varepsilon)^{-۱}\widetilde{\mathbf{h}}_۲ \qquad (۴۹-۳)$$

و با توجه به اینکه

$$(E_\varepsilon^T E_\varepsilon)^{-۱} = \mathrm{diag}(\underbrace{\varepsilon^{-۲},\varepsilon^{-۲},\ldots,\varepsilon^{-۲}}_{m_۱},\underbrace{۱,۱,\ldots,۱}_{M+۱},\underbrace{\varepsilon^{-۲},\ldots,\varepsilon^{-۲}}_{m_۲})$$

و بر اساس لم ۱۶.۳ ، داریم

$$A^T E_\varepsilon^T E_\varepsilon \widehat{\mathbf{h}}_{۲,\varepsilon} = ۰ \ . \qquad (۵۰-۳)$$

اگر دو رابطهٔ (۳-۴۷) و (۳-۵۰) را با هم مقایسه کنیم، ملاحظه می‌شود که بردارهای $\widehat{\mathbf{h}}_{۲,\varepsilon}$ و $A\Lambda_\varepsilon - \mathbf{e}$ هردو در فضای پوچی $A^T E_\varepsilon^T E_\varepsilon$ که یک بُعدی است، واقع می‌شوند. بنابراین اسکالر حقیقی $\widehat{\alpha}_{۲,\varepsilon}$ موجود است که

$$A\Lambda_\varepsilon - \mathbf{e} = \widehat{\alpha}_{۲,\varepsilon}\widehat{\mathbf{h}}_{۲,\varepsilon} \ . \qquad (۵۱-۳)$$

با میل دادن $\varepsilon \to \infty$، دستگاه خطی زیر با مجهولات $\widehat{\alpha}_2$ و $\widehat{\Lambda}$، به عنوان دستگاه مشخصهٔ مسألهٔ حذف گره با تقریب L_2 گسسته بدست می آید:

$$A\widehat{\Lambda} - \mathbf{e} = \widehat{\alpha}_2 \widehat{\mathbf{h}}_2 \qquad (۵۲-۳)$$

که در آن بردار $\widehat{\mathbf{h}}_2 = [\widehat{h}_{\bar{\mu}-1} \cdots \widehat{h}_k]^T$ با مؤلفه‌های زیر داده می‌شود:

$$\widehat{h}_i = \begin{cases} (-1)^{k-i} \begin{bmatrix} k-1 \\ i+1 \end{bmatrix} , & \tilde{\mu} - 1 + m_1 \le i \le k - m_2 , \\ 0 , & \text{در غیر اینصورت} \end{cases} \qquad (۵۳-۳)$$

باز هم می‌توان دستگاه مزبور را به کمک قضیهٔ ۱۲.۳ به طور صریح حل کرد. در قضیهٔ زیر این نتایج به طور دقیق برای محاسبه ضرایب وزنی λ_i و تعیین کران بالای تابع خطا در روش حذف گره با تقریب L_2 گسسته، جمعبندی می‌شوند.

قضیه‌ی ۱۷.۳ اگر و تنها اگر ضرایب وزنی λ_i با رابطهٔ

$$\lambda_i = \begin{cases} 0 , & \tilde{\mu} - 1 \le i \le \tilde{\mu} - 2 + m_1 , \\ \widehat{\gamma}_2^{-1} \sum_{j=\bar{\mu}-1}^{i} \begin{bmatrix} k-1 \\ j+1 \end{bmatrix}^2 , & \tilde{\mu} - 1 + m_1 \le i \le k - 1 - m_2 , \\ 1 , & k - m_2 \le i \le k - 1 \end{cases} \qquad (۵۴-۳)$$

تعیین شوند، که در آن

$$\widehat{\gamma}_2 = \sum_{j=\bar{\mu}-1+m_1}^{k-m_2} \begin{bmatrix} k-1 \\ j+1 \end{bmatrix}^2 \qquad (۵۵-۳)$$

آنگاه برای چندضلعی کنترلی داده شدهٔ $\{\tilde{c}_i\}_{i=1}^{n+1}$ با $D \neq 0$، چندضلعی کنترلی $\{\bar{c}_i\}_{i=1}^{n}$ که طبق (۳-۱۶) بدست می آید، جواب مینیمم کنندهٔ $\|\tilde{\mathbf{c}} - A\bar{\mathbf{c}}\|_2$ با قید $\bar{Q} \equiv \tilde{Q}$ روی $\mathcal{I} \setminus \mathcal{H}_{m_1,m_2}$ است. به علاوه، داریم

$$\operatorname{dist}(\tilde{Q}, \bar{Q}) \le \|\tilde{\mathbf{c}} - A\bar{\mathbf{c}}\|_2 = \widehat{\gamma}_\infty^{-1/2} \|D\|_2 . \qquad (۵۶-۳)$$

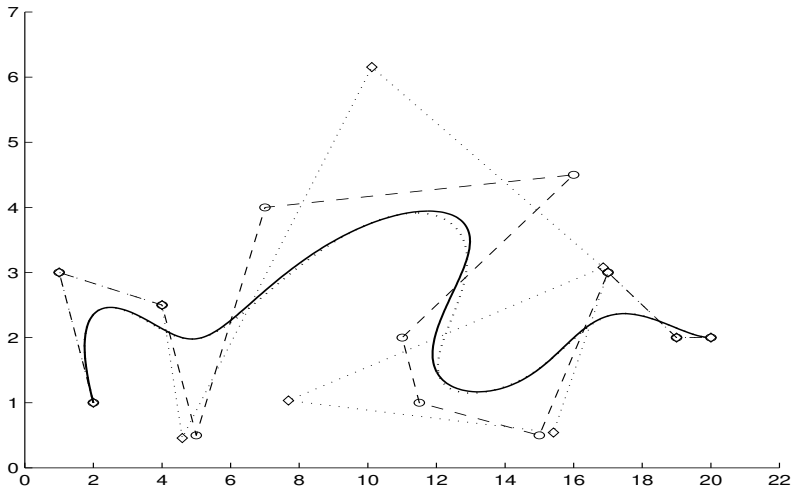
تبصره. منحنی $\bar{Q}$ بدست آمده از قضیهٔ ۱۷.۳ به ازای $m_1 = m_2 = 0$، جواب مسألهٔ تقریب L_2 گسسته برای دستگاه ناسازگار بزرگ $\tilde{\mathbf{c}}_L - A_L\bar{\mathbf{c}}_L$ نیز است. این موضوع مستقیماً با توجه به ساختار خاص A_L در (۲-۵) نتیجه می‌شود.

خطای منحنی ارائه شده در قضیهٔ ۱۷.۳ را می‌توان براساس (۳–۳۰) به طور دقیقتری تعیین کرد.

نتیجه‌ی ۱۸.۳ برای منحنی حذف گره شدهٔ $\bar{Q}$، ارائه شده در قضیهٔ ۱۷.۳، داریم

$$\operatorname{dist}(\tilde{Q},\bar{Q}) \le \widehat{\gamma}_{\text{۲}}^{-\text{۱}} \max\left\{ \begin{bmatrix} k-\text{۱} \\ j+\text{۱} \end{bmatrix} : \ \tilde{\mu}-\text{۱}+m_{\text{۱}} \le j \le k-m_{\text{۲}} \right\} \|D\|_{\text{۲}} \ . \qquad (\text{۵۷–۳})$$

یک محاسبهٔ ساده نشان می‌دهد که کران داده‌شده در (۳–۵۷) بهتر از کران داده‌شده در (۳–۵۶) است. امّا هنوز کران تعیین شده به وسیلهٔ جواب مسألهٔ تقریب L_∞ گسسته در (۳–۴۴) کوچکترین مقدار خطای بدست آمده برای تقریبهای گسسته می‌باشد.



شکل ۳–۷: حذف گره با تقریب $L_{\text{۲}}$ گسسته.

مثال ۱۹.۳ در شکل ۳–۷، حذف گره با تقریب $L_{\text{۲}}$ گسسته و با همان مفروضات مثال ۱۵.۳ انجام شده‌است. کران بدست آمده برای خطای بین دو منحنی ظریف و درشت در این روش طبق رابطهٔ(۳–۵۷) برابر است با ۱٫۹۳۹۰ ، در حالیکه $\widehat{\gamma}_{\text{۲}}^{-\text{۱}}\|D\|_{\text{۲}} = \text{۱٫۹۳۹۳}$. △

۳–۳–۴ تقریب L_∞ پیوسته

در این بخش مستقیماً برای یافتن منحنی مینیمم‌کنندهٔ $\mathrm{dist}(\tilde{Q},\bar{Q})$، اقدام می‌کنیم. فرض کنید R_{min} نشاندهندهٔ مینیمم R در (۳–۳۰) باشد. بدین ترتیب اختلاف نقطه‌وار بین منحنی بهترین تقریب پیوسته $\bar{Q}(t)$ و منحنی اولیه $\tilde{Q}(t)$ به صورت

$$\varrho(t) = \tilde{Q}(t) - \bar{Q}(t) = h(t).R_{min}.D\ , \qquad t \in \mathcal{H}_{m_1,m_2} \tag{۵۸–۳}$$

نوشته می‌شود، که در آن تابع معْین $h(t)$ به صورت

$$h(t) = \sum_{j=\bar{\mu}-1+m_1}^{k-m_2} h_j \tilde{B}^k_{j+J}(t) = \tilde{\mathbf{B}}^T \mathbf{h}_\infty \tag{۵۹–۳}$$

با ضرایب (مجهول) $h_j \in I\!R$ معرفی می‌شود و بردار (مجهول) $\mathbf{h}_\infty$ به صورت

$$\mathbf{h}_\infty = [\underbrace{0\,0\cdots0}_{m_1}\ \underbrace{h_{\bar{\mu}-1+m_1}\cdots h_{k-m_2}}_{M+1}\ \underbrace{0\cdots0\,0}_{m_2}]^T \tag{۶۰–۳}$$

در نظر گرفته می‌شود. این تابع معین باید در شرط

$$\max\{|h(t)|\ :\ t \in \mathcal{H}_{m_1,m_2}\} = 1 \tag{۶۱–۳}$$

صدق کند. بدین ترتیب با توجه به (۳–۵۸) و (۳–۵۹) و بر اساس لم ۱۰.۳ ، داریم

$$\mathbf{e} - A\Lambda = R_{min}\mathbf{h}_\infty\ . \tag{۶۲–۳}$$

بازهم با بکارگیری قضیهٔ ۱۲.۳ در مورد دستگاه (۳–۶۲) ، ضرایب وزنی λ_i بدست خواهند آمد. بنابراین تنها کافیست ضرایب Bـاسپلاین تابع $h(t)$ را بدست آوریم. متأسفانه در اینجا نمی‌توان عبارات صریحی برای h_i ها ارائه کرد.

با اینحال از دیدگاه نظری می‌توان شرایط مشخصه‌ای برای تابع خطای $\varrho(t)$ معرفی کرد. این مشخصه توسط **هندزکام** در [۲۳] برای بررسی یک مسأله مشابه مسألهٔ حذف گره مورد بحث در اینجا، ارائه شده است.

قضیه‌ی ۲۰.۳ تابع خطای تغییر یافتهٔ زیر را در نظر بگیرید:

$$\widetilde{\varrho}(t) = \begin{cases} \tilde{Q}(t) - \bar{Q}(t)\ , & t \in (\tilde{u}_{j_1}\ \tilde{u}_{j_2}) \\ \circ\ , & \text{در غیر اینصورت} \end{cases} \qquad (۶۳-۳)$$

که $\tilde{Q}(t), \bar{Q}(t) \subset I\!R^1$ اسپلاین Bـی و توابع $1 \leq j_1 < n_2 < j_2 \leq n+k+1$ و $j_2 - j_1 \geq k$ طبق (۳-۱۸) تعریف شده‌اند. آنگاه برای $k \geq 3$، تابع $\bar{Q}_\circ(t)$ جواب یکتای مینیمم‌سازی $\max\{|\widetilde{\varrho}(t)| : t \in [\tilde{u}_1\ \tilde{u}_{n+k+1}]\}$ است اگر و تنها اگر $\widehat{\varrho}_\circ(t)$ (تابع خطای تغییر یافته متناظر با مینیمم کننده؛ $\bar{Q}_\circ(t)$) دارای $j_2 - j_1 - k + 1$ نقطهٔ اکسترمال تناوبی مثل $\{e_i\}_{i=1}^{j_2-j_1-k+1}$ باشد، به طوری که

$$\tilde{u}_{j_1} < e_1 < e_2 < \cdots < e_{j_2-j_1-k+1} < \tilde{u}_{j_2} \qquad (۶۴-۳)$$

و

$$\widehat{\varrho}_\circ(e_i) = (-1)^{k-i}.\max\{|\widehat{\varrho}_\circ(t)| : t \in [\tilde{u}_1\ \tilde{u}_{n+k+1}]\}\ . \qquad (۶۵-۳)$$

تبصره.

۱) شرایط قضیهٔ ۲۰.۳ خیلی کلی‌تر از شرایط موضوع مورد بحث ماست، از آنجا که $(u_{j_1}\ u_{j_2}) \supseteq \mathcal{H}_{m_1,m_2}$.

۲) توجه کنید که Bـاسپلاینهای از مرتبهٔ $k \leq 2$ مشمول قضیهٔ ۲۰.۳ نمی‌شوند، اگر چه حالت $k = 2$ بعداً به طور جداگانه بررسی خواهد شد.

۳) **بلاتر و شوماکر** [۴] و **نورنبرگر و سامر** [۳۱] مسألهٔ یافتن بهترین تقریب یک تابع $f \in C[a\ b]$ بر روی فضای اسپلاینها $\mathcal{S}^k_{n,U}$ را به طور کامل و دقیق، ولی از دیدگاه نظری مورد بررسی قرار داده‌اند، و شرایط دقیقی برای یکتایی چنین تقریبی بدست آورده‌اند. به بیان ساده، در آنجا ثابت می‌شود که وقتی و تنها وقتی $n \leq 2k$ ، f دارای بهترین تقریب منحصربفرد در $\mathcal{S}^k_{n,U}$ است. برقرار کردن این شرایط برای حصول این یکتایی در مورد اسپلاینها ضروری است، چون فضای اسپلاینها یک فضای چبیشف ضعیف است و در حالت کلی ضرورتاً بهترین تقریب پیوستهٔ یک تابع در این فضا یکتا نیست. در ادامه نشان داده می‌شود که این شرایط را به طور کامل برای مسألهٔ مورد بحث در اینجا

نمی‌توان برقرار نمود، با این‌حال قضیهٔ ۲۰.۳ شرط ضعیفتری برای یکتایی بهترین تقریب پیوستهٔ تابع خاص $f = \tilde{Q} \in \mathcal{S}^k_{n+1,\bar{\mathcal{U}}}$ بر روی $\mathcal{S}^k_{n,\mathcal{U}}$ تعیین می‌کند.

برای بررسی شرایط مورد بحث در تبصرهٔ اخیر، مشاهده می‌شود که تابع $\varrho(t)$ را نیز می‌توان به صورت تابع خطای بین یک تابع پایهٔ Bـاسپلاین از فضای $\mathcal{S}^k_{n+1,\bar{\mathcal{U}}}$ و یک زیرفضای $\mathcal{S}^k_{n,\mathcal{U}}$ در نظر گرفت:

$$\varrho(t) = \left(\tilde{B}^k_{n_2}(t) - \sum_{j=n_1+m_1}^{n_2-1-m_2} \widetilde{\lambda}_{j-J}.B^k_j(t) \right).D\,, \qquad t \in \mathcal{H}_{m_1,m_2}\,.$$

به عبارت دیگر، در اینجا می‌خواهیم بهترین تقریب تابع $\tilde{B}^k_{n_2}(t)$ را بر روی زیرفضای تولید شده توسط $\widehat{\mathcal{B}} = \{B^k_j(t)\}_{j=n_1+m_1}^{n_2-1-m_2}$ بدست آوریم. نتایج بدست آمده در [۴] و [۳۱] از این حیث قابل بکارگیری در اینجا نیستند، چون $\widehat{\mathcal{B}}$ اساساً یک پایه برای تمام اسپلاینهای تعریف شده روی $\mathcal{H}_{m_1,m_2}$ نیست. بناچار موضوع را در جهت برقراری قضیهٔ ۲۰.۳ پی می‌گیریم.

بنابراین تنها با ارضاء شرایط قضیهٔ ۲۰.۳، می‌توان یکتایی $h(t)$ را تضمین کرد. برای محاسبهٔ $h(t)$ با بکارگیری قضیهٔ ۲۰.۳ برای $(u_{j_1}\ u_{j_2}) = \mathcal{H}_{m_1,m_2}$، ما باید تابع $h(t)$ در (۳–۵۹) را به گونه‌ای جستجو کنیم که در $M+1$ نقطهٔ $\{e_i\}_{i=\tilde{\mu}-1+m_1}^{k-m_2}$ به طور تناوبی به ماکسیمم و مینیمم خود، یعنی، $+1, -1$ برسد. به عبارت دیگر، برای

$$\tilde{u}_{n_1+m_1} < e_{\tilde{\mu}-1+m_1} < e_{\tilde{\mu}+m_1} < \cdots < e_{k-m_2} < \tilde{u}_{n_2+k-m_2} \qquad (۳–۶۶)$$

داشته باشیم

$$h(e_i) = (-1)^{k-i}\,, \qquad \tilde{\mu}-1+m_1 \le i \le k-m_2\,. \qquad (۳–۶۷)$$

مسألهٔ یافتن این نقاط تناوبی برای بهترین تقریب پیوسته، توسط رده‌ای از الگوریتمها موسوم به **الگوریتم رمز**[۷] حل می‌شود. هرچند برخی روشهای دیگر مانند روشهای برنامه‌ریزی خطی نیز برای حل آن وجود دارد. الگوریتم رمز که گاه روش تعویضی[۸] نیز نامیده می‌شود، از دیدگاه کلی بدین ترتیب انجام

[۷] Remez یا Remes

[۸] Exchange method

می‌شود که ابتدا یک حدس اولیه برای نقاط تناوبی را می‌گیرد، سپس به طور تکراری، مقدار خطا را در این نقاط درونیابی می‌کند و برخی از نقاط تناوبی را با نقاط اکسترمالِ این تابعِ درونیاب، به طور مناسب جایگزین می‌کند، تا به برخی شرایط همگرایی برسد.

الگوریتمی که در اینجا استفاده می‌شود، یک حالت ویژه از الگوریتم رمز نوع دوم است که برای مسألهٔ حذف گره در Bـاسپلاینها، به طور خاص تنظیم شده است. این الگوریتم به طور کلی به صورت زیر بیان می‌شود و در فصل آینده به طور مفصل دربارهٔ جزئیات پیاده‌سازی آن بحث خواهد شد.

۱) قرار بده $r = ۰$ و برای $i = \tilde{\mu} - ۱ + m_۱, \ldots, k - m_۲$ محاسبه کن:

$$e_i^{[۰]} = \tilde{\xi}_{i+J} = \frac{۱}{k-۱} \sum_{j=i+۱}^{i+k-۱} \tilde{u}_{j+J} \ . \qquad (۶۸-۳)$$

۲) تابع معین $h^{[r]}(t)$ را به نحوی تعیین کن که

$$h^{[r]}(e_i^{[r]}) = (-۱)^{k-i} \ , \qquad i = \tilde{\mu} - ۱ + m_۱, \ldots, k - m_۲ \ . \qquad (۶۹-۳)$$

۳) $e_i^{[r+۱]}$ را به عنوان نقاط اکسترمال تابع $h^{[r]}(t)$ محاسبه کن، به طوری که برای $i = \tilde{\mu} - ۱ + m_۱, \ldots, k - m_۲$ داشته باشیم $\text{sign}\left(h^{[r]}(e_i^{[r+۱]})\right) = (-۱)^{k-i}$.

۴) اگر $\max\{|h^{[r]}(t)| : t \in \mathcal{H}_{m_۱,m_۲}\}$ بقدر کافی به ۱ نزدیک نیست، قرار بده $r = r + ۱$ و به گام ۲ برگرد.

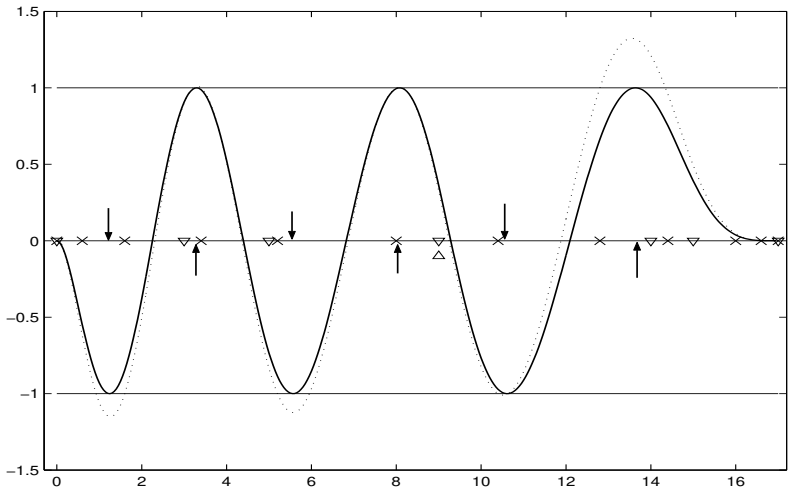
تبصره.

۱) اگر به تعریف تابع $h(t)$ توجه کنید، ملاحظه می‌شود که آن در اصل خطای بهترین تقریب پیوستهٔ یک تابع پایهٔ $\tilde{B}^k_{i,n+۱}$ بر روی فضای $\mathcal{S}^k_{n,\mathcal{U}}$ است. این موضوع مشابه تعریف چندجمله‌ای چبیشف به عنوان خطای بهترین تقریب t^n به وسیلهٔ چندجمله‌ایهای از درجهٔ $n - ۱$ روی $[۰ \ ۱]$ است. بنابراین، اصطلاح اسپلاین چبیشفی[۹] برای نامگذاری $h(t)$ مناسب به نظر می‌رسد.

[۹] Chebyshevian spline

۲) یک ویژگی مهم که در [۳۲] ثابت می‌شود، نشان می‌دهد که نقاط تناوبی $e_i^{[r]}$ بدست آمده در هر تکرار الگوریتم رمز در شرط زیر قرار می‌گیرند:

$$\tilde{u}_{i+J+1} < e_i^{[r]} < \tilde{u}_{i+J+k-1}\ , \quad r \geq 0,\ \ i = \tilde{\mu} - 1 + m_1, \ldots, k - m_2\ . \qquad (۳-۷۰)$$



شکل ۳—۸: یک اسپلاین چبیشفی مرتبهٔ ٦ بدست آمده از الگوریتم رمز: $h^{[0]}(t)$ (منحنی نقطه‌چین)، $h^{[4]}(t)$ (منحنی توپر)، گره‌ها (مثلثهای کوچک)، نقاط گرویل (ضربدر) و نقاط اکسترمال تناوبی نهایی با علامت فلش مشخص شده‌اند.

مثال ۲۱.۳ در شکل ۳—۸، نتیجه یک اجرای الگوریتم رمز در رابطه با حذف گره‌ای مشابه مثال ۱۵.۳، آمده است که بعد از ۴ تکرار در حد دقت اعشاریک کامپیوتر به جواب همگرا شده است. ضرایب h_j بدست آمده در این مثال عبارتند از:

$$h_1 = -۴/۱۸۰۱, \quad h_2 = ۹/۴۴۹۸, \quad h_3 = -۸/۴٦٦۱,$$

$$h_4 = ۸/۰۵۰۲, \quad h_5 = -٦/۷۱٦۳, \quad h_٦ = ۳/۹۴۰۰$$

△

با اجرای عددی الگوریتم رمز، ضرایب h_j مورد نیاز بدست می‌آیند. بدین ترتیب، با بکارگیری مجدد قضیهٔ ۱۲.۳ می‌توان ضرایب وزنی λ_i و این‌بار مقدار دقیق خطا در روش حذف گره با بهترین تقریب L_∞ پیوسته را بدست آورد. این نتایج در قضیهٔ زیر خلاصه می‌شوند.

قضیه‌ی ۲۲.۳ اگر و تنها اگر ضرایب وزنی λ_i با رابطهٔ

$$\lambda_i = \begin{cases} ۰ , & \tilde{\mu} - ۱ \le i \le \tilde{\mu} - ۲ + m_۱ , \\ \gamma_\infty^{-۱} \sum_{j=\bar{\mu}-۱}^{i} \begin{bmatrix} k-۱ \\ j+۱ \end{bmatrix} |h_j| , & \tilde{\mu} - ۱ + m_۱ \le i \le k - ۱ - m_۲ , \\ ۱ , & k - m_۲ \le i \le k - ۱ \end{cases} \qquad (۷۱-۳)$$

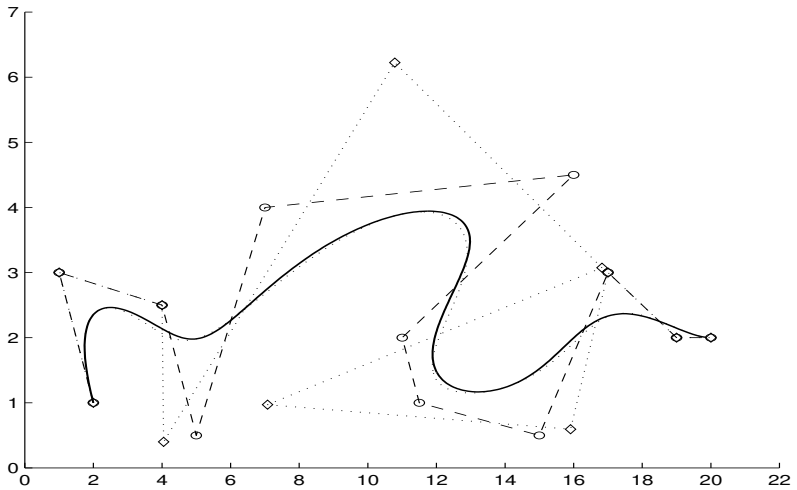
تعیین شوند، که در آن

$$\gamma_\infty = \sum_{j=\bar{\mu}-۱+m_۱}^{k-m_۲} \begin{bmatrix} k-۱ \\ j+۱ \end{bmatrix} |h_j| \qquad (۷۲-۳)$$

آنگاه برای $k \ge ۳$ و چندضلعی کنترلی داده شدهٔ $\{\tilde{c}_i\}_{i=۱}^{n+۱}$ با $D \neq ۰$، چندضلعی کنترلی $\{\bar{c}_i\}_{i=۱}^{n}$ که طبق (۳–۱۶) بدست می آید، جواب مینیمم‌کنندهٔ $\operatorname{dist}(\tilde{Q}, \bar{Q})$ با قید $\tilde{Q} \equiv \bar{Q}$ روی $\mathcal{I} \setminus \mathcal{H}_{m_۱, m_۲}$ است. به علاوه، داریم

$$\operatorname{dist}(\tilde{Q}, \bar{Q}) = \gamma_\infty^{-۱} \, \|D\|_۲ \, . \qquad (۷۳-۳)$$

در حالت $k = ۲$، منحنی Bـاسپلاین دقیقاً بر چندضلعی کنترلی خود منطبق است. بدین ترتیب، مسألهٔ حذف گره با بهترین تقریب L_∞ پیوسته به مسألهٔ حذف گره با تقریب L_∞ گسسته، تقلیل می‌یابد.



شکل ۳–۹: حذف گره با تقریب L_∞ پیوسته.

مثال ۲۳.۳ در شکل ۳–۹، حذف گره با تقریب L_∞ پیوسته و با همان مفروضات مثال ۱۵.۳ انجام شده‌است. خطای بین دو منحنی ظریف و درشت در این روش برابر $\gamma_\infty^{-۱}\|D\|_۲ = ۰/۱۹۹۱$ است. $\triangle$

این فصل را با ذکر این نکته به پایان می‌رسانیم که ضرایب h_j بدست آمده در هر سه تقریب مزبور، شرط نتیجهٔ ۱۳.۳ را صدق می‌دهند و بنابراین $۰ \leq \lambda_{\bar{\mu}-۱} \leq \lambda_{\bar{\mu}} \leq \cdots \leq \lambda_{k-۲} \leq \lambda_{k-۱} \leq ۱$.

تناوبی بودن علامت ضرایب $\widehat{h}_i$ در تقریبهای گسسته بر اساس تعریفشان در (۳–۴۱) و (۳–۵۳) مشخص است. در مورد h_j بدست آمده از الگوریتم رمز، در همین‌جا می‌توان گفت که چون منحنی $h(t)$ در M نقطه تغییر علامت می‌دهد، بر اساس خاصیت کاهش نوسان در Bـاسپلاینها، ضرایب آن یعنی h_j نیز لااقل باید در همین تعداد از مؤلفه‌ها تغییر علامت دهد.

فصل ۴

پیاده‌سازی

۴–۱ پیاده‌سازی الگوریتم رمز

در فصل پیش مسألهٔ حذف گره از دیدگاههای گوناگون مورد ارزیابی قرار گرفت و سه روش حذف گره با تقریبهای L_∞ گسسته، L_2 گسسته و L_∞ پیوسته ارائه شد. بیائید یک بار دیگر روند کلی یک الگوریتم حذف گره را آنگونه که در فصل پیش ذکر شد، مطالعه کنیم. این الگوریتم را می‌توان با الگویی به صورت زیر در نظر گرفت:

۱) منحنی Bـاسپلاین $\tilde{Q}$ با چندضلعی کنترلی $\{\tilde{c}_i\}_{i=1}^{n+1}$ از مرتبهٔ k روی دنبالهٔ گره‌های $\tilde{U} = U \cup \{t\}$ داده شده است. با حذف گره $t = \tilde{u}_{\tilde{\delta}}$ می‌خواهیم منحنی Bـاسپلاین $\bar{Q}$ با چندضلعی کنترلی $\{\bar{c}_i\}_{i=1}^{n}$ روی دنبالهٔ گره‌های U را چنان بدست آوریم که دارای کمترین $\mathcal{E}$ـاختلاف با منحنی $\tilde{Q}$ باشد. منظور از $\mathcal{E}$ یکی از سه نرم L_∞ گسسته، L_2 گسسته یا L_∞ پیوسته است.

۲) حذف گره مجازی. چندضلعیهای $\{c_i^I\}_{i=1}^{n}$ و $\{c_i^{II}\}_{i=1}^{n}$ را بر اساس روابط (۳–۱۴) و (۳–۱۵) محاسبه کن.

۳) حل مسألهٔ تقریب. با توجه به $\mathcal{E}$، ضرایب h_j یا $\widehat{h}_j$ را بدست آور و ضرایب وزنی λ_i را به وسیلهٔ روابط (۳–۴۲) یا (۳–۵۴) یا (۳–۷۱) محاسبه کن.

۴) ترکیب گرانیگاهی. چندضلعی کنترلی $\{\bar{c}_i\}_{i=1}^n$ را به عنوان نتیجهٔ الگوریتم، مطابق رابطهٔ (۳–۱۶) به صورت ترکیب وزنی $\{c_i^I\}_{i=1}^n$ و $\{c_i^{II}\}_{i=1}^n$ با ضرایب λ_i محاسبه کن.

در پیاده‌سازی عملی الگوریتم فوق، در گام ۳ برای محاسبهٔ ضرایب $\widehat{h}_j$ وقتی که تقریب، روی نرمهای گسستهٔ L_∞ و L_2 در نظر گرفته می‌شود، روابط (۳–۴۱) و (۳–۵۳) به طور صریح در اختیار است. امّا متأسفانه برای محاسبهٔ ضرایب h_j در تقریب L_∞ پیوسته، باید از روشهای تکراری همگرا به جواب مانند الگوریتم رمز استفاده کرد. نگارنده علیرغم جستجوی زیاد در مراجع، راه‌حلهای دقیقی برای رفع ابهامات در پیاده‌سازی عملی این الگوریتم نیافت، هرچند مطالعات بسیار گسترده‌ای از دیدگاه نظری دربارهٔ این الگوریتم صورت گرفته‌است [۳۷]، [۳۱] و [۳۲].

بنابراین بیائید الگوریتم رمز را به طور دقیقتر نگاه کنیم تا ببینیم چه ابهاماتی ممکن است در پیاده‌سازی عملی آن رخ دهد.

در گام اول برای شروع روند تکرار، نقاط گرویل $\tilde{\xi}_i$ به عنوان یک حدس اولیه برای نقاط اکسترمال تناوبی $e_i^{[0]}$ انتخاب می‌شوند. این نقاط از این جهت برای شروع مناسب هستند که چندضلعی دِبور پارامتری شدهٔ هر تابع Bـاسپلاین $h^{[r]}(t)$ در این نقاط از رئوس خود و در اصل به خاطر تناوبی بودن ضرایب $h_j^{[r]}$، از اکسترممهای خود می‌گذرد. چون هر تابع Bـاسپلاین یک تقریب نزدیک به چندضلعی دِبور خود است، بنابراین به طور شهودی می‌توان انتظار داشت که تابع Bـاسپلاین $h^{[r]}(t)$ نیز در این نقاط به اکسترممهای خود نزدیک باشد.

در گام دوم، تابع Bـاسپلاین $h^{[r]}(t)$ را باید به گونه‌ای بیابیم که به طور تناوبی مقادیر $+1, -1$ را روی نقاط $e_i^{[r]}$ درونیابی کند. برای این کار کافیست یک دستگاه معادلات خطی با ابعاد $(M+1)\times(M+1)$ را حل کنیم. با توجه به (۳–۶۹) و نمادگذاری (۳–۵۹) ، این دستگاه به صورت

$$\sum_{j=\tilde{\mu}-1+m_1}^{k-m_2} h_j^{[r]}\,\tilde{B}_{j+J}^k(e_i^{[r]}) = (-1)^{k-i}\,, \qquad i=\tilde{\mu}-1+m_1,\ldots,k-m_2 \qquad (۴–۱)$$

بدست می‌آید. در اینجا $h_j^{[r]}$ مجهولات هستند و برای محاسبهٔ ضرایب این دستگاه $\tilde{B}_{j+J}^k(e_i^{[r]})$، می‌توان روابط بازگشتی (۱–۱) و (۱–۲) را به ازای $M+1$ مقدار $e_{\tilde{\mu}-1+m_1}^{[r]},\ldots,e_{k-m_2}^{[r]}$ ارزیابی کرد.

با توجه به ویژگی (۳–۷۰) در مورد مکان قرارگیری $e_i^{[r]}$ ها در دنبالهٔ گره‌ها، مشاهده می‌شود که دستگاه (۴–۱) نوعاً یک ماتریس پرصفر و نواری شکل است و اگر m_1, m_2 مطابق (۳–۱۹) به طور مناسب انتخاب شده باشند، این ماتریس نامنفرد است و درونیابی امکانپذیر خواهد بود.

ابهام کار در گام سوم الگوریتم رمز پیش می‌آید. در این گام باید نقاط اکسترمال تابع $h^{[r]}(t)$ محاسبه شوند و به جای نقاط تناوبی قبلی جایگزین گردند. حل ریشه‌های مشتق $h'^{[r]}(t) = 0$ به عنوان اولین راه حل به فکر می‌رسد. استفاده از این روش حقیقتاً از دیدگاه نظری معتبر است چون تمام نقاط اکسترمال درون بازهٔ $\mathcal{H}_{m_1,m_2}$ قرار دارند (رابطهٔ(۳–٦٦) را ملاحظه کنید) و با چشم‌پوشی از حالاتی که $h^{[r]}(t)$ دارای پیوستگی C^1 نیست، طبق مباحث مقدماتی می‌توان گفت که ریشه‌های مشتق بر نقاط اکسترمال تابع منطبق است.

برای حل ریشه‌های مشتق، درگیر شدن با معادلهٔ اسپلاینی $h'^{[r]}(t) = 0$ به طور صریح، به خاطر صورت عمدتاً غیرخطی و پیچیدهٔ آن در عمل جوابگو نخواهد بود، هر چند در مورد Bـاسپلاین مربعی $k = 3$، ریشه‌های مشتق به صورت ریشه‌های یک تابع قطعه‌وار خطی به سادگی قابل حل است. در عوض روشهای تکرار برای بدست آوردن نقاط اکسترمال تابع برای پیاده‌سازی عملی تر هستند. در این روشها به طور کلی در هر مرحله یک نقطهٔ تکراری را در مسیر نزدیک شدن به نقطهٔ اکسترمال حرکت می‌دهیم. محاسبهٔ دقیق نقاط اکسترمال توسط این روشها مستلزم تکرارهای زیاد برای همگرایی به جواب است و اگر از این روشها در الگوریتم رمز بدین ترتیب استفاده شود، با تکرارهای تو در تو مواجه می‌شویم که به طور اساسی باعث کند شدن اجرای الگوریتم می‌شود.

برای رفع تمام موانع مذکور، ما **یک** تکرار از روش نیوتن با طول گام **یک** را در هر تکرار الگوریتم رمز بکار خواهیم برد. روش نیوتن برای محاسبهٔ نقطهٔ اکسترمال تابع $f \in C^2[a\ b]$ به صورت زیر بیان می‌شود:

۱) یک نقطهٔ آغازی $t_0 \in [a\ b]$ را انتخاب کن.

۲) تا رسیدن به برخی شرایط همگرایی تکرار کن:

$$t_{r+1} = t_r - \omega \frac{f'(t_r)}{f''(t_r)}, \qquad r = 0, 1, \ldots$$

که $\omega > 0$ طول گام است و در هر تکرار به طور مناسب انتخاب می‌شود.

این الگوریتم برای هر تابع اکیداً محدب $f'' > 0$ (اکیداً مقعر $f'' < 0$) و با شروع از هر نقطهٔ آغازی در سراسر دامنه $[a\ b]$ به مینیمم (ماکسیمم) مطلق f همگرا می‌شود. در شرایط دیگر با انتخاب t_0 به طوری که $f''(t_0) > 0$ و تکرار الگوریتم در نواحی که $f''(t) > 0$ است نیز می‌توان به مینییم موضعی f همگرا شد (به طور مشابه ماکسیمم). به علاوه، با انتخاب t_0 در نزدیکی جواب، روش نیوتن با سرعت همگرایی مرتبه دو به نقطهٔ اکسترمال نزدیک می‌شود و این، استفاده از روش نیوتن در الگوریتم رمز را بسیار مطلوب می‌کند.

مشابه روش کلاسیک نیوتن، ما طول گام را $\omega = 1$ انتخاب می‌کنیم و به علاوه، تنها یک تکرار روش نیوتن را در هر مرحله بکار می‌بریم. این ساده‌سازی علاوه بر اینکه باعث کاهش محاسبات در اجرای کلی الگوریتم رمز می‌شود، با توجه به ویژگیهای خاص Bـاسپلاینها و با در نظر گرفتن برخی ملاحظات، اعتبار همگرایی روش نیوتن را نیز در بردارد.

یکی از مزایای الگوریتم رمز تنظیم شده برای Bـاسپلاینها در این است که همگرایی این الگوریتم چندان به انتخاب اکسترمالهای دقیق $h^{[r]}(t)$ به جای نقاط تناوبی قبلی $e_i^{[r]}$، در گام ۳ حساس نیست، بلکه با جایگزین کردن یک اکسترمال تقریبی با $e_i^{[r]}$ نیز می‌توان به جواب همگرا شد. به عبارت دیگر، تنها کافیست در هر تکرار، هر نقطهٔ $e_i^{[r+1]}$ به گونه‌ای انتخاب شود که مقدار تابع $h^{[r]}$ روی آن، کمی از نوار عرضی بین دو خط $+1, -1$ خارج شود. بنابراین گام ۳ الگوریتم رمز را به صورت زیر تغییر می‌دهیم:

$e_i^{[r+1]}$ را بعنوان اکسترمالهای تقریبی تابع $h^{[r]}(t)$ محاسبه کن، به طوری که برای $i = \tilde{\mu} - 1 + m_1, \ldots, k - m_2$ داشته باشیم $\operatorname{sign}\left(h^{[r]}(e_i^{[r+1]}) - (-1)^{k-i}\right) = (-1)^{k-i}$.

این عدم حساسیت بدلیل خاصیت کاهش نوسان در Bـاسپلاینهاست و بواسطهٔ همین عدم حساسیت، یک تکرار نیوتن برای نزدیک شدن به نقطهٔ اکسترمال و برقراری شرط فوق در هر تکرار الگوریتم رمز کافیست.

توجه داشته باشید که اگر در هر تکرار، در گام ۳ مقادیر $h^{[r]}(e_i^{[r+1]})$ به قدر کافی از نوار عرضی $+1, -1$ فاصله نداشته باشند، آنگاه روند همگرایی الگوریتم رمز کند خواهد شد. با انتخاب $\omega = 1$ به نوعی این تضمین حاصل می‌شود که چنین وضعیتی پیش نیاید. با این حال برای بکارگیری روش نیوتن روی تابعی که به طور سراسری اکیداً محدب یا اکیداً مقعر نیست، مثل تابع نوسانی $h^{[r]}(t)$، طول گام باید به

گونه‌ای انتخاب شود که شرط

$$h''^{[r]}(t) > 0 \quad , \qquad \text{حول نقطهٔ مینیمم} \tag{۴—۲}$$

$$h''^{[r]}(t) < 0 \quad , \qquad \text{حول نقطهٔ ماکسیمم} \tag{۴—۳}$$

در حین تکرارهای نیوتن ثابت باقی بماند. جستجوی ω مناسب برای برقراری شرایط (۴—۲) و (۴—۳) در هر تکرار، نیاز به محاسباتی اضافی دارد و باعث کند شدن اجرای الگوریتم می‌شود. به همین دلیل است که مقدار ثابت ۱ را برای طول گام قرار می‌دهیم و در عوض با افزودن یک مرحلهٔ تصحیحی در الگوریتم، شرایط فوق را در تکرارهای متوالی حفظ می‌کنیم.

در ابتدا به نظر می‌رسید که چون در نزدیکی نقاط اکسترمال شرایط (۴—۲) و (۴—۳) برقرار هستند، نیازی به ارزیابی آنها در هر تکرار نیست ولی حقیقت آنست که در مواردی در حین برخی تکرارها، طول گام ۱ برای نزدیک شدن به نقطهٔ اکسترمال زیاد است و $e_i^{[r+1]}$ از آن دور می‌شود، به قدری که جهت تحدب $h^{[r]}(t)$ در آن تغییر می‌یابد. به طور کلی، در اطراف $e_i^{[r]}$ با توجه به علامت $(-1)^{k-i}$، ما بدنبال مینیمم وقتی $k-i$ فرد است یا ماکسیمم وقتی $k-i$ زوج است، می‌گردیم. بدین ترتیب شرایط (۴—۲) و (۴—۳) به صورت $(-1)^{k-i}h''^{[r]}(t) > 0$ خلاصه می‌شوند. با در نظر گرفتن دو شرط ضروری دیگر طبق (۳—۶۶)، برای مواقعی که نقاط $e_i^{[r]}$ خیلی از اکسترمالهای دقیق دور می‌شوند، برقراری هر یک از روابط زیر را نشاندهندهٔ دور شدن $e_i^{[r]}$ از اکسترمم دقیق می‌دانیم:

$$e_i^{[r]} > e_{i+1}^{[r]} \simeq e_{i+1}^{[r-1]} \qquad \text{یا} \qquad e_i^{[r]} < e_{i-1}^{[r]} \tag{۴—۴}$$

$$(-1)^{k-i}h''^{[r]}(e_i^{[r]}) < 0 \qquad \text{یا} \tag{۴—۵}$$

در صورت برقراری هریک از روابط فوق روش نیوتن را کنار می‌گذاریم و مرحلهٔ تصحیحی را به کار می‌بندیم. روش تصحیحی بکار گرفته شده در اینجا مبتنی بر خواص Bـاسپلاینهاست و از یک دیدگاه هندسی بیان می‌شود.

نکتهٔ آخر اینکه برای شروع یک تکرار نیوتن در مرحلهٔ r ام الگوریتم رمز، به طور منطقی نقاط $e_i^{[r]}$ را انتخاب می‌کنیم. انتخاب نقاط $\tilde{\xi}_i$ در آغاز کار و $e_i^{[r]}$ در هر تکرار الگوریتم رمز به عنوان نقاط آغازین تکرار نیوتن به واسطهٔ نزدیکی به اکسترمالهای دقیق مناسب به نظر می‌رسد و همگرایی مرتبه دو مطلوب در هر

مرحله را تضمین می‌کند. بدین ترتیب با جمعبندی مطالب مذکور، الگوریتم رمز به صورت زیر پیاده‌سازی می‌شود:

۱) قرار بده $r = ۰$ و برای $i = \tilde{\mu} - ۱ + m_۱, \ldots, k - m_۲$ بگذار $e_i^{[۰]} = \tilde{\xi}_{i+J}$.

۲) ضرایب $h_j^{[r]}$ را با حل دستگاه (۴–۱) بدست آور.

۳) برای هر $i = \tilde{\mu} - ۱ + m_۱, \ldots, k - m_۲$، اگر $\tilde{\mu}(e_i^{[r]}) < k - ۱$ آنگاه اعمال زیر را انجام بده

(a) « روش نیوتن». قرار بده

$$\widehat{e}_i \;=\; e_i^{[r]} - \frac{h'^{[r]}(e_i^{[r]})}{h''^{[r]}(e_i^{[r]})}\,. \qquad (۶–۴)$$

(b) اگر $(-۱)^{k-i} h''^{[r]}(e_i^{[r]}) < ۰$ و $e_{i-۱}^{[r+۱]} < \widehat{e}_i < e_{i+۱}^{[r]}$ آنگاه قرار بده $e_i^{[r+۱]} = \widehat{e}_i$، در غیر اینصورت

« روش تصحیحی»

$$e_i^{[r+۱]} \;=\; \tilde{\xi}_{i+J} + \frac{۱}{۲}\big((-۱)^{k-i} - h_i^{[r]}\big)\Big(\frac{۱}{d_i^{[r]}} + \frac{۱}{d_{i+۱}^{[r]}}\Big), \qquad (۷–۴)$$

که $\{d_j^{[r]}\}_{j=\tilde{\mu}-۱+m_۱}^{k+۱-m_۲}$ ضرایب Bـاسپلاین مشتق $h'^{[r]}(t)$ هستند.

۴) اگر $\max\{|h^{[r]}(t)| \,:\, t \in \mathcal{H}_{m_۱,m_۲}\}$ به قدر کافی به ۱ نزدیک نیست، قرار بده $r = r + ۱$ و به گام ۲ برگرد.

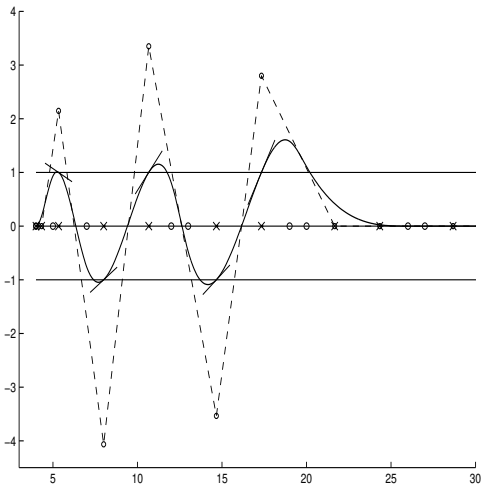
در صورت بروز هر یک از شرایط (۴–۲) یا (۴–۳) در تکرار r ام الگوریتم رمز، روش تصحیحی از اطلاعات بدست آمده در تکرارهای قبل به یک تعبیر هندسی استفاده می‌کند. فرض کنید خط $y = ۱$ تابع Bـاسپلاین $h^{[r]}(t)$ را در دو نقطهٔ A و B در دو طرف نقطهٔ ماکسیمم C قطع کند (به تکرار دوم در شکل ۴–۱ نگاه کنید). به وضوح داریم، یا $A_x = e_i^{[r]}$ یا $B_x = e_i^{[r]}$. اگر نقاط E و F محل تقاطع این خط با چندضلعی دِبور باشند، به واسطهٔ خاصیت کاهش نوسان در Bـاسپلاینها به سادگی می‌توان نشان داد که $E_x < C_x < F_x$. بنابراین طول نقطهٔ وسط پاره‌خط EF یک انتخاب مناسب برای $e_i^{[r]}$ می‌تواند باشد. همین استدلال وقتی که $e_i^{[r]}$ را می‌خواهیم به مینیمم تابع نزدیک کنیم نیز با بررسی نقاط تقاطع تابع و

چندضلعی دِبور آن با خط $y = -1$ وجود دارد. روابط محاسبهٔ این نقطه به طور ساده شده در (۴–۷) بکار گرفته شده است.

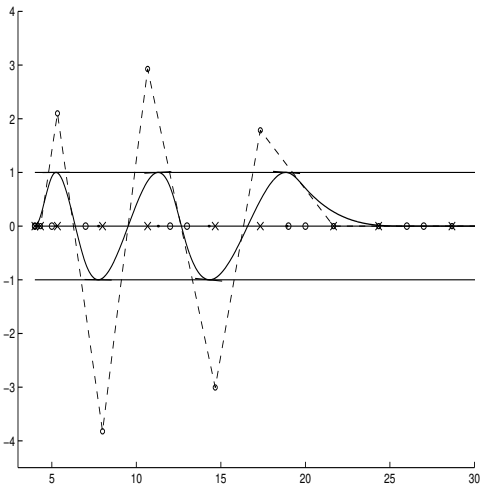
مثال ۱.۴ در شکل ۴–۱، برای بدست آوردن اسپلاین چبیشفی مرتبهٔ ۴ مرتبط با حذف گره $\tilde{u}_7 = ۱۲$ از دنباله گره‌های $\tilde{U} = \{۴, ۴, ۴, ۴, ۵, ۷, ۱۲, ۱۳, ۱۹, ۲۰, ۲۶, ۲۷, ۳۳, ۳۳, ۳۳, ۳۳\}$، پنج تکرار متوالی الگوریتم رمز نمایش داده شده‌است. در این مثال در تکرار دوم، یک مرحلهٔ تصحیحی صورت می‌گیرد. مقایسهٔ مینیمم‌ها و ماکسیمم‌های $h^{[r]}(t)$ در بین تکرارها، سرعت همگرایی این الگوریتم را به طور شهودی نشان می‌دهد. △

تبصره.

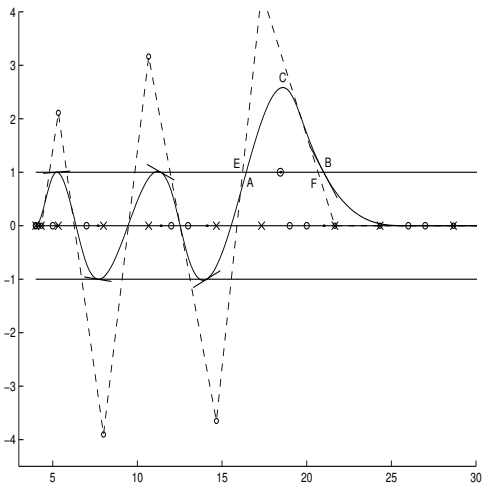
۱) اگر $h'^{[r]}(t)$ در $e_i^{[r]}$ پیوسته نباشد به وضوح اعتبار روش نیوتن از بین می‌رود. برای یک Bـاسپلاین از مرتبهٔ $k > ۲$، این وضعیت تنها موقعی بوجود می‌آید که $e_i^{[r]}$ یک گره با مرتبهٔ تکرار $k - ۱$ باشد. امّا جالب است بدانید که این وضعیت نه تنها مزاحم نیست، بلکه کاملاً مطلوب نیز است. به طور دقیق می‌توان نشان داد که در این وضعیت $e_i^{[r]} = \tilde{\xi}_{i+J}$ در تمام تکرارهای الگوریتم رمز به عنوان یک نقطهٔ اکسترمال تناوبی، ثابت باقی می‌ماند. این بدان دلیل است که هر تابع $h^{[r]}(t)$ از یک طرف به واسطهٔ تکرر $k - ۱$ مرتبه‌ای $e_i^{[r]}$، ضریب کنترلی $h_i^{[r]}$ را در $e_i^{[r]}$ درونیابی می‌کند و از طرف دیگر بعد از گام ۲ مقدار $(-۱)^{k-i}$ را در همان نقطه درونیابی می‌کند. بدین ترتیب دو نقطهٔ $\begin{pmatrix} \tilde{\xi}_{i+J} \\ h_i^{[r]} \end{pmatrix}$ و $\begin{pmatrix} e_i^{[r]} \\ (-۱)^{k-i} \end{pmatrix}$ بر هم منطبق می‌شوند. امّا می‌دانیم که نقطهٔ اول اکسترمم چندضلعی دِبور پارامتری شده است. بنابراین بواسطهٔ خاصیت پوشش محدب در Bـاسپلاینها، این نقطه اکسترمم تابع $h^{[r]}(t)$ نیز است. یک نمونه از این وضعیت در شکل ۴–۲ نشان داده شده‌است. در این شکل، تکرارهای الگوریتم رمز تا همگرایی به اسپلاین چیبشفی مرتبهٔ ۴ که در رابطه با حذف گره $\tilde{u}_8 = ۷$ از دنباله گره‌های $\tilde{U} = \{۰, ۰, ۰, ۰, ۳, ۴, ۵, ۷, ۱۱, ۱۱, ۱۱, ۱۵, ۱۷, ۱۹, ۱۹, ۱۹\}$ مطرح می‌شود، نمایش داده شده‌است.



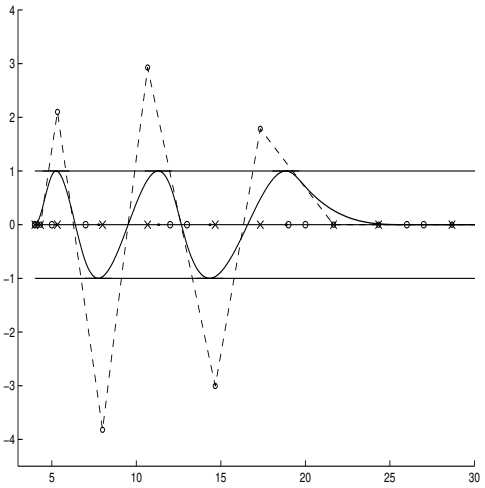
1) max=1.6075 , min=-1.0894



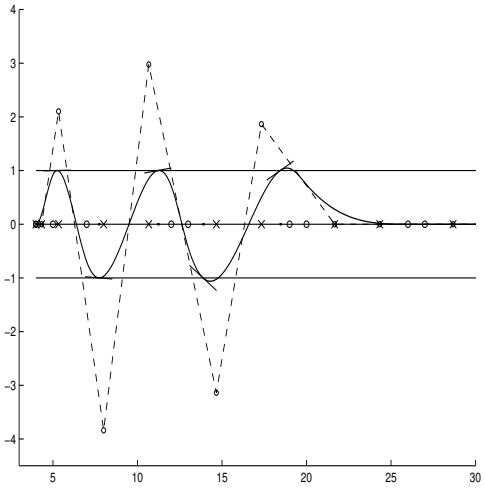
4) max=1.0004 , min=-1.0007



2) max=2.5831 , min=-1.0209



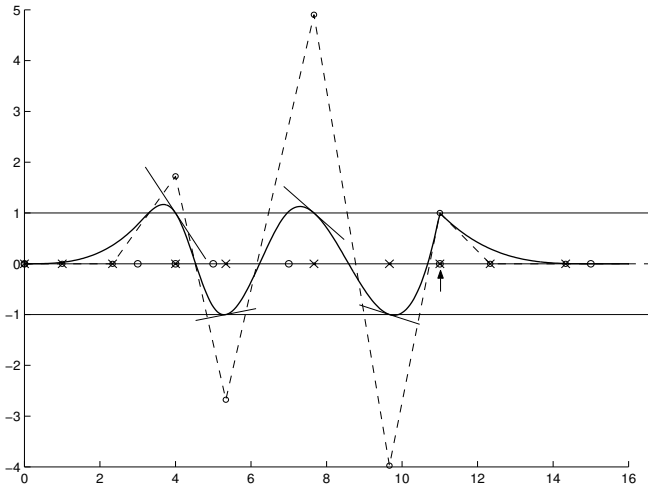
5) max=1.0000 , min=-1.0000



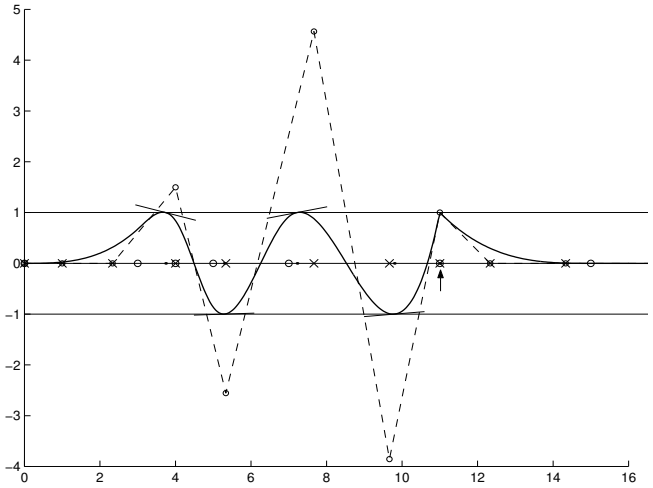
3) max=1.0458 , min=-1.0620

در هر شکل مثلثهای کوچک مکان گره‌ها، ضربدرها مکان نقاط گرویل، نقاط توپر مکان نقاط $e_i^{[r]}$ و پاره‌خطها، شیب مماس بر منحنی در نقاط $e_i^{[r]}$ را نشان می‌دهند. توجه کنید که در تکرار دوم مکان نقطهٔ $e_5^{[2]}$ با نقطهٔ وسط EF تصحیح می‌شود.

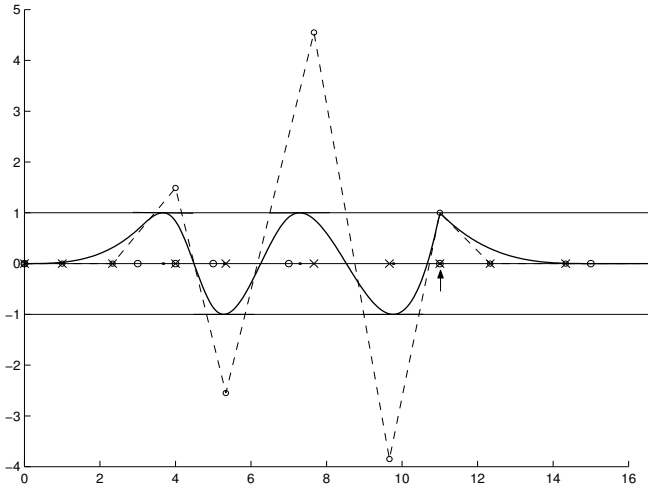
شکل ۴–۱: تکرارهای الگوریتم رمز پیاده‌سازی شده.



1) max=1.1665 , min=-1.0142



2) max=1.0067 , min=-1.0009



3) max=1.0000 , min=-1.0000

شکل ۴–۲: تکرارهای الگوریتم رمز وقتی تابع $h^{[r]}(t)$ روی یک گره مشتقپذیر نیست: بعد از اولین درونیابی مکان نقطهٔ $e_5^{[r]}$ ثابت می‌ماند.

۲) حالاتی که مشتق دوم $h''^{[r]}(t)$ در برخی نقاط $e_i^{[r]}$ ناپیوسته است باعث بدحالت شدن محاسبهٔ (۴–۶) می‌شود. امّا این وضعیت نیز در تکرار بعدی الگوریتم با گذر از مرحلهٔ تصحیحی اصلاح می‌شود.

۳) با یک بار بکارگیری روابط بازگشتی (۱–۱) و (۱–۲) برای هر $e_i^{[r]}$، مقادیر کلیهٔ توابع Bـاسپلاین در جدول مثلثی زیر را می‌توان محاسبه کرد:

$$
\begin{array}{ccccccccccc}
 & & & & & & & & \vdots & & \\
 & & & & & & & & \circ & & \\
 & & & & & & & & \vdots & & \\
 & & & & & & & & \circ & & \tilde{B}^{k}_{\delta_i-k+1}(e_i^{[r]}) \\
 & & & & & & \vdots & & & \nearrow & \\
 & & & & & & \circ & & \tilde{B}^{k-1}_{\delta_i-k+2}(e_i^{[r]}) & \to & \tilde{B}^{k}_{\delta_i-k+2}(e_i^{[r]}) \\
 & & & & & \nearrow & & \nearrow & & \nearrow & \\
 & & & & \tilde{B}^{k-2}_{\delta_i-k+3}(e_i^{[r]}) & \to & \tilde{B}^{k-1}_{\delta_i-k+3}(e_i^{[r]}) & \to & \tilde{B}^{k}_{\delta_i-k+3}(e_i^{[r]}) & & \\
\vdots & & & \nearrow & & \nearrow & & \nearrow & & & \\
\circ & & \cdots & & \vdots & & \vdots & & \vdots & & \\
 & \nearrow & & \nearrow & & \nearrow & & \nearrow & & & \\
\tilde{B}^{1}_{\delta_i}(e_i^{[r]}) & \to & \cdots & \to & \tilde{B}^{k-2}_{\delta_i}(e_i^{[r]}) & \to & \tilde{B}^{k-1}_{\delta_i}(e_i^{[r]}) & \to & \tilde{B}^{k}_{\delta_i}(e_i^{[r]}) & & \\
\circ & & & & \circ & & \circ & & \circ & & \\
\vdots & & & & \vdots & & \vdots & & \vdots & &
\end{array}
\qquad (۸–۴)
$$

که $\delta_i = \tilde{\delta}(e_i^{[r]})$ اندیس مکان جایگیری $e_i^{[r]}$ در $\tilde{\mathcal{U}}$ است. به خصوص با ذخیرهٔ سه ستون آخر این جدول مقادیر پایهٔ Bـاسپلاین مورد نیاز برای حل دستگاه در روال درونیابی (گام ۲) و نیز مشتقات اول و دوم مورد نیاز در روال تکرار نیوتن (گام ۳) را در اختیار داریم. این کار به طور مؤثر از انجام محاسبات اضافی جلوگیری و الگوریتم را سریعتر می‌کند.

پیاده‌سازی الگوریتم رمز به صورت مذکور در عمل بسیار کارا، دقیق و سریع خواهد بود. بکارگیری این الگوریتم به ازای مقادیر مختلف k و دنبالهٔ گره‌های متفاوت نشان می‌دهد که تقریباً با k تکرار الگوریتم رمز به شکل فوق، می‌توان با دقت قابل قبولی به جواب که همان اسپلاین چبیشفی است، نزدیک شد. بنابراین ما عمدتاً به جای تکرار الگوریتم تا حصول شرط همگرایی، به طور ثابت k تکرار الگوریتم را انجام می‌دهیم. البته، منطقی است اگر، تکرارهایی را که در آنها الگوریتم یک یا چندبار وارد مرحلهٔ تصحیحی می‌شود، به حساب نیاوریم.

در هر تکرار از الگوریتم رمز به طور میانگین $\mathcal{O}(k^3)$ محاسبهٔ اصلی انجام می‌شود. اگر حدود k تکرار الگوریتم را برای رسیدن به جواب کافی فرض کنیم، آنگاه محاسبهٔ ضرایب اسپلاین چبیشفی به طور میانگین $\mathcal{O}(k^4)$ محاسبهٔ اصلی نیاز دارد.

۴–۲ روشهای ترکیب و تجزیه

الگوریتمهای حذف تک گره، که تاکنون دربارهٔ آنها بحث شد، هنگامی در کاربردهای عملی مفید خواهند بود که به وسیلهٔ آنها بتوان تعداد زیادی از گره‌ها را به طریق منطقی حذف کرد. هرچند برخی کاربردهای ویژه وجود دارند که دقیقاً یک اجرای الگوریتم حذف تک گره را بکار می‌گیرند (مثل مسألهٔ رتبه‌بندی گره‌ها)، عموماً مایلیم با بکارگیری متوالی این الگوریتم بر روی یک منحنی یا رویه، با کاهش داده‌ها آن را با تعداد کمتری رئوس کنترلی تقریب بزنیم.

برای در دست داشتن چنین روالهایی که قابلیت استفاده در طیف وسیعی از کاربردهای گوناگون را برای روشهای حذف تک گره فراهم کنند، در ادامه چند روش ترکیب و تجزیه[۱] مبتنی بر الگوریتم حذف تک گره تشریح می‌شوند. به بیان ساده، هدف از بکارگیری یک روش تجزیه، جداسازی اطلاعات یک شیء مدل شده (مثل منحنی، رویه، عکس و ...) به دو بخش کوچکتر، یکی داده‌های تقریب زننده و دیگری داده‌های خطا است. در ترکیب، برعکس، دو دسته از داده‌ها که یکی دربردارندهٔ تقریبی درشت از مدل و دیگری نشاندهندهٔ جزئیات مدل است، با یکدیگر ترکیب می‌شوند و شیء مورد نظر با دقت بالاتری نمایش می‌یابد.

اصل مبحث روشهای ترکیب و تجزیه از نظریهٔ موجکها[۲] برخاسته است و با مفهوم تفکیک پذیری چندگانه[۳] در ارتباط است. در اینجا بدون ورود به مباحث موجک چند روش ترکیب و تجزیه مشابه مفاهیم موجک بیان می‌شود. خوانندهٔ علاقه‌مند به موضوع موجکها می‌تواند به [٦] و [۲۹] مراجعه کند. امّا ایدهٔ بکارگیری الگوریتمهای حذف گره برای ترکیب و تجزیه به طور اساسی توسط **دَهلن و لیخ** در [۱۰] مورد توجه قرار گرفته است. آنها چندین روش ترکیب و تجزیه برای منحنیها، رویه‌های ضرب تانسوری،

[۱] Composition and Decomposition

[۲] Wavelets

[۳] Multiresulotion

رویه‌های با توپولوژی دلخواه و تصاویر، بر اساس یک روش حذف چند گره با تقریب L_2 گسسته ارائه کردند.

همانطور که در بخش ۳–۳ در فصل پیش اشاره شد، اگر منحنی $\tilde{Q}$ با رئوس کنترلی $\{\tilde{c}_i\}_{i=1}^{n+1}$ روی دنبالهٔ گره‌های $\tilde{\mathcal{U}}$ در دست باشد، با بکارگیری یک الگوریتم حذف گره طبق (۳–۳۹) داریم

$$\tilde{\mathbf{c}} = A\bar{\mathbf{c}} + \alpha \mathbf{h} D \qquad (۹–۴)$$

که بردارهای $\tilde{\mathbf{c}}$ و $\bar{\mathbf{c}}$ به ترتیب شامل مؤلفه‌هایی از رئوس کنترلی ظریف و درشت است که حین عمل حذف گره تغییر می‌کنند. عدد D ضریب خطای حذف گره و α و $\mathbf{h}$ وابسته به نوع تقریب در نظر گرفته شده، بر اساس یکی از روابط (۳–۴۱) یا (۳–۵۳) و یا توسط الگوریتم رمز تعیین می‌شوند. در اینجا رابطهٔ (۴–۹) اساس روشهای تجزیه و ترکیب قرار می‌گیرد.

ابتدا فرض کنید منحنی ظریف $\tilde{Q}$ با مشخصات فوق در اختیار است. با بکارگیری یک الگوریتم حذف گره داریم

$$\begin{array}{ccc} \tilde{\mathbf{c}} & \longrightarrow \bar{\mathbf{c}} & \qquad\qquad \tilde{\mathcal{U}} \longrightarrow \mathcal{U} \\ & \searrow & \qquad\qquad\quad \searrow \\ & D & \qquad\qquad\quad\ \tilde{u}_{\tilde{\delta}} \end{array} \qquad (۱۰–۴)$$

این شمای الگوریتم تجزیه بر اساس حذف تک گره است. اگر دقت کنید ملاحظه می‌شود که تعداد اطلاعات ظریف قبل از تجزیه با تعداد مجموع داده‌ها بعد از تجزیه برابر است و این یک اصل ضروری در روشهای ترکیب و تجزیه است. از طرف دیگر، $\bar{\mathbf{c}}$ یک تقریب خوب برای $\tilde{\mathbf{c}}$ و در اصل بهترین تقریب آن در برخی نرمها می‌باشد.

حال فرض کنید منحنی درشت $\bar{Q}$ با رئوس کنترلی $\{\bar{c}_i\}_{i=1}^{n}$ روی دنبالهٔ گره‌های $\mathcal{U}$ در اختیار است. اگر D و $t = \tilde{u}_{\tilde{\delta}}$ نیز در دست باشند، با بکارگیری (۴–۹) می‌توانیم $\tilde{Q}$ را بازسازی کنیم:

$$\begin{array}{ccc} \bar{\mathbf{c}} \longrightarrow & \tilde{\mathbf{c}} & \qquad\qquad \mathcal{U} \longrightarrow \tilde{\mathcal{U}} \\ & \nearrow & \qquad\qquad\quad \nearrow \\ D & & \qquad\qquad t \end{array} \qquad (۱۱–۴)$$

این شمای الگوریتم ترکیب را نشان می‌دهد. توجه کنید که اگر قرار دهیم $D = 0$، آنگاه این الگوریتم دقیقاً معادل الگوریتم درج تک گره خواهد شد.

حال ملاحظه می‌شود که الگوریتم تجزیه (۴–۱۰) را می‌توان با حذف پی در پی گره‌ها به طور متوالی روی داده‌های تقریب زننده تکرار کرد و بدین ترتیب تعداد داده‌های تقریب زننده را کمتر و داده‌های

اولیه را به داده‌های خطای بیشتری تجزیه نمود. اگر الگوریتم (۴–۱۰) را r بار تکرار کنیم، داریم

$$\begin{array}{ccccccccccc} \tilde{\mathbf{c}} = \bar{\mathbf{c}}_0 & \longrightarrow & \bar{\mathbf{c}}_1 & \longrightarrow & \bar{\mathbf{c}}_2 & \longrightarrow & \cdots & \longrightarrow & \bar{\mathbf{c}}_{r-1} & \longrightarrow & \bar{\mathbf{c}}_r \\ & \searrow & & \searrow & & & & & & \searrow & \\ & & D_1 & & D_2 & & & & & & D_r \end{array} \qquad (۱۲–۴)$$

$$\begin{array}{ccccccccccc} \tilde{\mathcal{U}} = \mathcal{U}_0 & \longrightarrow & \mathcal{U}_1 & \longrightarrow & \mathcal{U}_2 & \longrightarrow & \cdots & \longrightarrow & \mathcal{U}_{r-1} & \longrightarrow & \mathcal{U}_r \\ & \searrow & & \searrow & & & & & & \searrow & \\ & & t_1 & & t_2 & & & & & & t_r \end{array} \qquad (۱۳–۴)$$

که $t_i \in \mathcal{U}_{i-1}$ و در هر مرحله از تجزیه، یک اجرای الگوریتم حذف تک گره بکار گرفته می‌شود. بدین ترتیب، اگر داده‌های اولیه $\tilde{\mathbf{c}}$ شامل n مؤلفه باشد، بعد از r بار تجزیه، r دادهٔ خطا $D_1, \ldots, D_r$ و یک دادهٔ تقریب زننده $\bar{\mathbf{c}}_r$ شامل $n-r$ مؤلفه در دست است.

به طور مشابه، با بکارگیری متوالی الگوریتم (۴–۱۱) ، وقتی r دادهٔ خطا $D_1, \ldots, D_r$ و r گره $t_1, \ldots, t_r$ در دسترس باشند، می‌توان مجدداً منحنی اولیه را بازسازی کرد:

$$\begin{array}{ccccccccc} \bar{\mathbf{c}}_r & \longrightarrow & \bar{\mathbf{c}}_{r-1} & \longrightarrow & \bar{\mathbf{c}}_{r-2} & \longrightarrow & \cdots & \longrightarrow & \bar{\mathbf{c}}_1 \longrightarrow \bar{\mathbf{c}}_0 = \tilde{\mathbf{c}} \\ \nearrow & & \nearrow & & & & & & \nearrow \\ D_r & & D_{r-1} & & & & & & D_1 \end{array} \qquad (۱۴–۴)$$

$$\begin{array}{ccccccccc} \mathcal{U}_r & \longrightarrow & \mathcal{U}_{r-1} & \longrightarrow & \mathcal{U}_{r-2} & \longrightarrow & \cdots & \longrightarrow & \mathcal{U}_1 \longrightarrow \mathcal{U}_0 = \tilde{\mathcal{U}} \\ \nearrow & & \nearrow & & & & & & \nearrow \\ t_r & & t_{r-1} & & & & & & t_1 \end{array} \qquad (۱۵–۴)$$

در شمای فوق داده‌ها در هر مرحله با رابطهٔ زیر بهم مربوط می‌شوند:

$$\begin{aligned} \bar{\mathbf{c}}_{i-1} &= A_i \bar{\mathbf{c}}_i + \alpha_i \mathbf{h}_i D_i \\ \mathcal{U}_{i-1} &= \mathcal{U}_i \cup \{t_i\}, \qquad i = r, \ldots, 1 \end{aligned} \qquad (۱۶–۴)$$

که A_i ماتریس درج گره t_i در $\mathcal{U}_{i-1}$ است و بردار $\mathbf{h}_i$ و ضریب α_i به طور مناسب بر اساس حل یک مسألهٔ تقریب در برخی نرمها بدست می‌آیند. آنچه لازم به تذکر است اینست که نرم بکارگرفته شده در مراحل بازسازی باید با نرم بکار رفته در فرایند تجزیه یکسان باشد.

مسألهٔ اساسی در پیاده‌سازی الگوریتمهای (۴–۱۲) و (۴–۱۴) چگونگی تعیین گره‌هایی است که در هر مرحله، حذف یا درج می‌شوند. روشهای متعددی را می‌توان برای انتخاب یک گره از دنبالهٔ گره‌ها $\mathcal{U}_i$ پیشنهاد کرد. به وضوح یک انتخاب آرمانی آنچنان است که گره حذف شده باعث بروز کمترین خطا در بین تمام گره‌های دیگر باشد. برای این کار باید خطای حذف هر یک از گره‌ها را محاسبه و کوچکترین در بین آنها را انتخاب کرد.

آنچنانکه در فصل پیش اثبات شد، به طور کلی خطای حذف گره را می‌توان به صورت $\gamma^{-1}\|D\|_2$ تخمین زد، که ضریب γ براساس حل یک مسألهٔ تقریب بدست می‌آید. اگر از نرمهای برداری L_∞ گسسته یا L_2 گسسته استفاده کنیم، بدون انجام محاسبات طولانی و بدون بکارگیری روابط حذف گره برروی رئوس کنترلی، می‌توانیم $\gamma = \widehat{\gamma}_\infty$ یا $\gamma = \widehat{\gamma}_2$ را بر اساس (۳–۴۳) یا (۳–۵۵) محاسبه کنیم. در این روابط تنها کافیست ضرایب براکتی متناظر با هر گره را بدست آوریم. امّا اگر از نرم تابعی L_∞ پیوسته در مرحلهٔ تقریب حذف گره استفاده کنیم، محاسبهٔ $\gamma = \gamma_\infty$ طبق (۳–۷۲) مستلزم محاسبهٔ ضرایب اسپلاین چبیشفی و اجرای الگوریتم رمز است و وقتی این عمل برای تمام گره‌ها به کار رود، زیر روال تعیین بهترین گره برای حذف را بسیار طولانی می‌کند.

بنابراین عمدتاً ترجیح می‌دهیم که از محاسبهٔ γ صرفنظر کنیم و تنها از $\|D\|_2$ برای مقایسهٔ خطای حذف هر گره استفاده کنیم. این ساده‌سازی از این حیث منطقی به نظر می‌رسد که برای یک دنبالهٔ گره یکنواخت، ضرایب براکتی متناظر با هر گره یکسانند و در نتیجه $\widehat{\gamma}_\infty$ و همچنین $\widehat{\gamma}_2$ برای تمام گره‌ها برابرند. به وضوح در این حالت تمام توابع اسپلاین چبیشفی نیز بجز یک انتقال با یکدیگر برابرند و در نتیجه دارای ضرایب یکسانند و بدین ترتیب γ_∞ نیز برای تمام گره‌ها مقدار یکسان دارد. این نتایج در مواردی که دنبالهٔ گره‌ها تقریباً متساوی‌الفاصله است نیز به طور تقریبی برقرار است. پس با انتخاب گره $t \in \mathcal{U}_i$ بقسمی که

$$\|D_t\|_2 = \min\left\{\|D_{u_j}\|_2 \,:\, u_j \in \mathcal{U}_i\right\}. \qquad (۴–۱۷)$$

می‌توان انتظار داشت که حذف t در بین سایر گره‌ها دارای کمترین خطا باشد. در اینجا منظور از D_t ضریب خطای حذف گره t است. توجه کنید که این انتخاب برای هر نوع حذف گره‌ای می‌تواند مناسب باشد، چون عامل $\|D\|_2$ در بین اندازهٔ خطای هر یک از تقریبهای مذکور مشترک است.

با این حال در برخی موارد که نیاز به دقت بیشتری در انتخاب بهترین گره برای حذف داریم، با صرف اندکی محاسبات بیشتر، گره t را به گونه‌ای انتخاب می‌کنیم که حذف آن دارای کمترین خطای L_∞ گسسته در بین سایر گره‌ها باشد:

$$\widehat{\gamma}_\infty^{-1}(t)\|D_t\|_2 = \min\left\{\widehat{\gamma}_\infty^{-1}(u_j)\|D_{u_j}\|_2 \,:\, u_j \in \mathcal{U}_i\right\}. \qquad (۴–۱۸)$$

در اینجا، $\widehat{\gamma}_\infty(t)$ بر اساس رابطهٔ (۳–۴۳) به ازای حذف گره t بدست می‌آید.

بعد از حذف اولین گره‌ای که بر اساس (۴–۱۷) یا (۴–۱۸) انتخاب شده است، دو روش کلی را می‌توان برای انتخاب گره‌های بعدی برای حذف در نظر گرفت. در روش اول، بعد از حذف اولین گره t_1 مجدداً روابط (۴–۱۷) یا (۴–۱۸) را بر روی دنبالهٔ گره‌های جدید $\mathcal{U}_1 = \mathcal{U}_0 \setminus \{t_1\}$ ارزیابی می‌کنیم و t_2 را به عنوان بهترین گره برای حذف در بین گره‌های $\mathcal{U}_1$ انتخاب می‌کنیم. به همین ترتیب گره‌های $t_3, t_4, \ldots$ را با ارزیابی مجدد ضرایب خطا در دنباله گره‌های $\mathcal{U}_2, \mathcal{U}_3, \ldots$ انتخاب می‌کنیم. در روش دیگر، تنها با یک بار ارزیابی روابط (۴–۱۷) یا (۴–۱۸) روی دنباله گره‌های اولیه $\tilde{\mathcal{U}}$، گره‌ها را به ترتیب صعودی برحسب $\|D_{\tilde{u}_j}\|_2$ (یا $(\widehat{\gamma}_\infty^{-1}(\tilde{u}_j)\|D_{\tilde{u}_j}\|_2$) در یک لیست مرتب می‌کنیم و بدون ارزیابی مجدد ضرایب خطا در هر مرحله، گره‌ها را به ترتیبی که در این لیست مرتب‌شده ظاهر می‌شوند، حذف می‌کنیم.

این دو روش کلی برای انتخاب گره‌ها برای حذف، اساس پیاده‌سازی روشهای ترکیب و تجزیهٔ مبتنی بر حذف تک گره قرار می‌گیرند. روش اول را انتخاب گره با ارزیابی مجدد و روش دوم را انتخاب گره با یک بار مرتب‌سازی می‌نامیم. هر دوی این روشها در اصطلاحات تحلیل الگوریتمها به عنوان یک الگوریتم سودجویانه یا حریص[۴] شناخته می‌شوند. این گونه الگوریتمها لزوماً در همهٔ موارد بهترین جواب در بین مجموعه جوابها را بدست نمی‌دهند، بلکه در هر مرحله یک انتخاب را که به طور موضعی خوب به نظر می‌رسد، برمی‌گزینند. در مقابل روشهای مبتنی بر الگوریتمهای حذف چند گره با بررسی خطای کلی بعد از حذف r گره سعی می‌کنند که بهترین گره‌ها را برای رسیدن به کمترین خطا پیدا کنند. در این باره کمی در بخش ۵–۱ صحبت خواهیم کرد.

روش اول در انتخاب گره یا روش با ارزیابی مجدد علیرغم صرف محاسبات مکرر، این مزیت نسبی را داراست که خطای بین دو منحنی ظریفتر و درشت تر در هر مرحله از الگوریتم تجزیه، به تقریب، کمترین مقدار در بین انتخابهای دیگر برای گره حذف شده است. مزیت روش دوم یا روش انتخاب گره با یک بار مرتب‌سازی در اینست که ضرایب خطا را تنها یک بار ارزیابی می‌کند، ولی به اندازهٔ خطای بین منحنیها در میان دو مرحلهٔ متوالی الگوریتم تجزیه بی‌توجه است. با این حال این موضوع در عمل قابل اغماض است و نتایج بدست آمده از روش مرتب‌سازی از برخی جهات بهتر از نتایج بدست آمده از روش ارزیابی مجدد است. از آن جمله می‌توان به این مورد اشاره کرد که چون خطای حذف گره‌ها در این روش نسبت به منحنی

[۴] Greedy Algorithm

اولیه سنجیده می‌شوند، منحنی بدست آمده بعد از انجام حذفهای متوالی، نزدیکتر به منحنی اولیه باقی می‌ماند. امّا در روش ارزیابی مجدد، تجمع خطاهای کوچک بین منحنیهای متوالی، در مواردی که تعداد زیادی از گره‌ها حذف می‌شوند، به یک خطای بزرگ بین منحنی اولیه و نهایی منجر می‌شود.

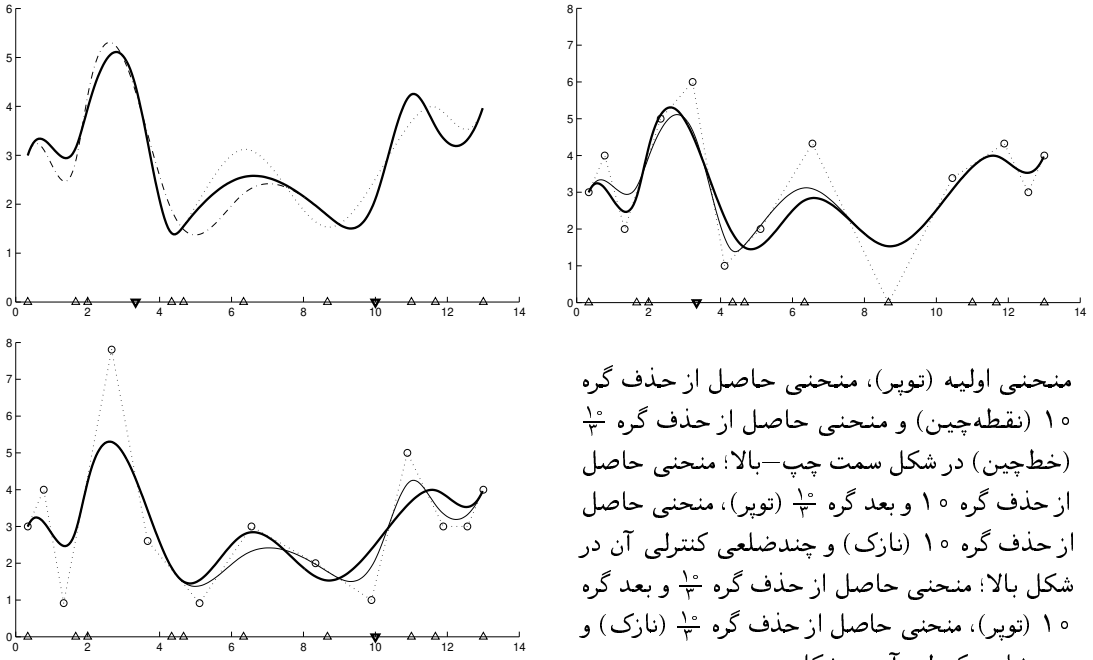
در عمل استفادهٔ متناوب از این دو روش برای انتخاب گره‌ها برای حذف به نتایج مطلوبتری هم از حیث صرفه‌جویی در محاسبات و هم از حیث نزدیک بودن تقریب نهایی به منحنی اولیه، منجر می‌شود. برای این کار، ابتدا با یک بار مرتب سازی گره‌ها تا چند مرحله در الگوریتم تجزیه پیش می‌رویم. سپس با ارزیابی مجدد ضرایب حذف گره نسبت به گره‌های باقی مانده، یک لیست مرتب‌شدهٔ جدید می‌سازیم و دوباره تا چند مرحله در الگوریتم تجزیه پیش می‌رویم و به همین ترتیب تا آخر.

یک نکته دیگر در رابطه با الگوریتمهای ترکیب و تجزیه لازم به تذکر است. یک بررسی ساده در روابط درج یک گره نشان می‌دهد که دو عمل درج متوالی با گره‌های t_1 و t_2 که $|\delta(t_1) - \delta(t_2)| \geq k$ ، مستقل از هم و به عبارتی جابجایی است. به طور مشابه، دربارهٔ حذف گره‌ها می‌توان نشان داد که دو عمل حذف متوالی با گره‌های t_1 و t_2 که $|\delta(t_1) - \delta(t_2)| \geq k+1$ ، نیز از یکدیگر مستقل و جابجایی است (به عنوان مثال به شکل ۴—۳ نگاه کنید). این موضوع بسیاری از محاسبات تکراری در روشهای ترکیب و تجزیه را کاهش می‌دهد. مثلاً اگر از روش ارزیابی مجدد، برای انتخاب گره‌ها برای حذف استفاده کنیم، در می‌یابیم که در هر مرحله تنها کافیست مقادیر D_{u_j} را به ازای گره‌های $\{u_j \in \mathcal{U}_i \ : \ |j - \delta(t_{i-1})| \leq k\}$ ارزیابی کنیم و از مقادیر محاسبه شده در مراحل قبلی به طور مناسب به جای مابقی مقادیر D_{u_j} در مرحله جدید استفاده کنیم. بدین ترتیب ملاحظه می‌شود که به میزان قابل توجهی از محاسبات کاسته می‌شود.

در برخی موارد مطلوبست که دنبالهٔ گره‌های حذف شده درالگوریتم تجزیه، به مفهومی که در بالا ذکر شد، نسبت بهم مستقل باشند. در این موارد نحوهٔ انتخاب گره‌ها برای حذف را به صورت زیر تغییر می‌دهیم:

$$\|D_{t_{i+1}}\|_2 = \min \left\{ \|D_{u_j}\|_2 \ : \ u_j \in \mathcal{U}_i \text{ و } |\delta_{\mathcal{U}_{i-1}}(t_i) - \delta_{\mathcal{U}_{i-1}}(u_j)| \geq k+1 \right\} . \qquad (۴—۱۹)$$

همین قید را می‌توانیم در رابطهٔ (۴—۱۸) نیز منظور کنیم. به علاوه، می‌توان روش استفادهٔ تناوبی از روشهای ارزیابی مجدد و مرتب‌سازی را با این نحوهٔ انتخاب گره‌ها ترکیب کرد و یک روش کارآمدتر بدست آورد.



منحنی اولیه (توپر)، منحنی حاصل از حذف گره ۱۰ (نقطه‌چین) و منحنی حاصل از حذف گره $\frac{10}{3}$ (خط‌چین) در شکل سمت چپ—بالا؛ منحنی حاصل از حذف گره ۱۰ و بعد گره $\frac{10}{3}$ (توپر)، منحنی حاصل از حذف گره ۱۰ (نازک) و چندضلعی کنترلی آن در شکل بالا؛ منحنی حاصل از حذف گره $\frac{10}{3}$ و بعد گره ۱۰ (توپر)، منحنی حاصل از حذف گره $\frac{10}{3}$ (نازک) و چندضلعی کنترلی آن در شکل روبرو.

شکل ۴—۳: حذف گره‌های مستقل: دو گره $\tilde{u}_7 = \frac{10}{3}$ و $\tilde{u}_{12} = 10$ از دنبالهٔ گره‌های $\{13, 13, 13, \frac{35}{3}, 13, 11, 10, \frac{26}{3}, \frac{19}{3}, \frac{14}{3}, \frac{13}{3}, \frac{10}{3}, 2, \frac{5}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{4}\}$ در یک منحنی B-اسپلاین مرتبهٔ ۴ به طور مستقل از هم حذف می‌شوند.

در پایان به طور اجمال معایب و مزایای روشهای ترکیب و تجزیه مبتنی بر حذف تک گره را نسبت به آنهایی که بر اساس حذف چند گره و مفاهیم موجک پیاده‌سازی می‌شوند، بیان می‌کنیم.

معایب

- در روشهای تجزیه مبتنی بر حذف چند گره، می‌توان با یک گام تعداد زیادی از گره‌ها را حذف کرد. همینطور در روشهای تجزیه مبتنی بر مفاهیم موجک، می‌توان تعداد رئوس کنترلی را به یک باره نصف کرد. در حالیکه انجام این کار در روش ما مستلزم تکرارهای متوالی الگوریتم حذف تک گره است، این برتری در مورد درج گره نیز وجود دارد.

- در روشهای تجزیه مبتنی بر حذف چند گره و نیز موجکها، خطای بین داده‌های اولیه و نهایی به طور مستقیم مورد ارزیابی قرار می‌گیرد و نتیجهٔ الگوریتم، یک مینیمم کنندهٔ خطا بین داده‌های اولیه و نهایی به ازای برخی نرمها است. در حالیکه در روش ما، اندازهٔ خطا مرحله به مرحله و بین دو منحنی متوالی در روند تجزیه ارزیابی می‌شود، و با آنکه بین هر دو مرحلهٔ متوالی خطا مینیمم است، امّا بعید

به نظر نمی‌رسد که تجمع این خطاها باعث بروز یک خطای فاحش بین داده‌های اولیه و نهایی گردد.

مزایا

- چون در روش ما حذف گره‌ها به صورت تدریجی و یک به یک صورت می‌گیرد، بدون یک برنامه‌ریزی کلی می‌توان همچنانکه در مراحل تجزیه پیش می‌رویم، به ارزیابی خطا بپردازیم و در صورت تجاوز آن از آستانهٔ قابل پذیرش، فرایند را متوقف کنیم، در حالیکه این عمل در روشهای مبتنی بر حذف چند گره و نیز موجکها نیاز به یک برنامه‌ریزی کلی و کنکاش در تعداد و چگونگی انتخاب گره‌های مناسب برای حذف دارد. در مورد این موضوع، در بخش ۵—۱، بحثهایی مطرح می‌شود.

- ویژگی تقریباً منحصربفرد روشهای تجزیه و ترکیب ارائه شده در اینجا، که به طور ذاتی به واسطهٔ بکارگیری حذف تک گره بدست می‌آید، در اینست که این روشها کاملاً مستقل از نوع دنبالهٔ گره‌های بکارگرفته شده برای مدل هستند و بر روی هر دنبالهٔ گره دلخواهی به خوبی عمل می‌کنند، در حالیکه بخصوص روشهای موجک و نیز پیاده‌سازیهای عملی روشهای حذف چند گره، عموماً بر روی دنباله گره‌های یکنواخت یا برخی حالات نزدیک به آن کار می‌کنند. البته، برخی تحقیقات بر روی موجکهای مبتنی بر دنبالهٔ گره‌های غیر یکنواخت در حال انجام است.

- مسألهٔ بهترین تقریب L_∞ پیوسته و محاسبهٔ ضرایب اسپلاین چبیشفی چه از لحاظ نظری، به واسطهٔ یکتا نبودن جواب آن، و چه از لحاظ عملی، به واسطهٔ پیچیدگی حل آن برای بدست آوردن یک جواب، در مورد حذف همزمان چند گره قابل بررسی نمی‌باشد. به همین دلایل، به ناچار در آنها نرمهای پیوسته را با برخی نرمهای گسسته تقریب می‌زنند. امّا در روشهای مبتنی بر حذف یک گره به وسیلهٔ الگوریتم کارآمد رمز می‌توان جواب یکتای آن را بدست آورد. حذف متوالی گره‌ها بر اساس بهترین تقریب L_∞ پیوسته هر چند لزوماً به یک جواب نهایی با کمترین خطای L_∞ پیوسته نمی‌رسد، امّا جواب نهایی به طور موضعی و به اندازهٔ چشمگیری به منحنی اولیه نزدیک است.

علاوه بر موارد مذکور، آنچه روشهای ترکیب و تجزیه معرفی شده در این رساله را در عمل با سایر روشها قابل رقابت می‌سازد، تنظیم الگوریتمهای سریع حذف تک گره و به خصوص الگوریتم رمز با کمترین

محاسبات و دقیقترین نتایج است.

۴–۳ نتایج

ابتدا به بررسی نتایج بدست آمده از اجراهای الگوریتم رمز می‌پردازیم. در بسیاری از موارد، ضرایب Bـاسپلاین تابع اسپلاین چبیشفی، برای حالت خاصی که حذف یک گره از دنباله گره‌های یکنواخت مد نظر است، مورد نیاز واقع می‌شوند. در این وضعیت، تفاوتی بین ضرایب h_j متناظر با حذف یک گره یا هر گره دیگر وجود ندارد (البته بجز گره‌های نزدیک به کناره‌های دامنه). بدین ترتیب با بکارگیری الگوریتم رمز دربارهٔ حذف هر تک گره‌ای از دنبالهٔ گره‌های یکنواخت نتایجی برابر با جدول ۴–۱ بدست می آید. نمودار اسپلاینهای چبیشفی متناظر با ضرایب هر یک از ستونهای این جدول در شکل ۴–۴ (ستون سمت چپ) آمده‌است. همانطور که در جدول ۴–۱ دیده می‌شود، ستونهای متناظر با k زوج متقارن و ستونهای متناظر با k فرد پادمتقارن هستند. بدین ترتیب اسپلاین چبیشفی متناظر با k زوج، یک تابع زوج و اسپلاین چبیشفی متناظر با k فرد، یک تابع فرد است.

مرتبهٔ اسپلاین ضرایب	$k=۲$	$k=۳$	$k=۴$	$k=۵$
$h_۰$	۱	−۱/۵۲۸۷۵۵۸۷	۱/۹۴۱۷۵۰۱۰	−۲/۴۲۸۵۰۱۱۵
$h_۱$	−۱	۱/۹۰۱۹۰۱۸۸	−۲/۶۸۰۱۵۳۴۸	۳/۸۱۸۱۰۱۸۷
$h_۲$	۱	−۱/۹۰۱۹۰۱۸۸	۲/۸۴۰۰۷۶۷۴	−۴/۳۳۱۱۲۰۰۰
$h_۳$	–	۱/۵۲۸۷۵۵۸۷	−۲/۶۸۰۱۵۳۴۸	۴/۳۳۱۱۲۰۰۰
$h_۴$	–	–	۱/۹۴۱۷۵۰۱۰	−۳/۸۱۸۱۰۱۸۷
$h_۵$	–	–	–	۲/۴۲۸۵۰۱۱۵

جدول ۴–۱: ضرایب اسپلاین چبیشفی متناظر با حذف تک گره از دنباله گره‌های یکنواخت.

مرتبهٔ اسپلاین ضرایب	$k=۲$	$k=۳$	$k=۴$	$k=۵$
$h_{k-۲}$	۱	۲/۴۱۴۲۱۳۵۶	۲/۸۱۴۵۱۱۸۷	۳/۰۱۱۱۴۴۲۱
$h_{k-۱}$	−۱	−۱	−۱	−۱
h_k	۱	۲/۴۱۴۲۱۳۵۶	۲/۸۱۴۵۱۱۸۷	۳/۰۱۱۱۴۴۲۱

جدول ۴–۲: ضرایب اسپلاین چبیشفی متناظر با حذف تک گره از دنباله گره‌های متساوی‌الفاصله با گره‌های $k-۱$ تکرار شده.

حالت پر کاربرد دیگر، حذف یک گره از دنبالهٔ گره‌های متساوی‌الفاصله‌ای است که هر گره آن دارای مرتبه تکرار $k-۱$ است. در این وضعیت نیز ضرایب h_j بدست آمده از حذف هر گره با گره‌های دیگر تفاوتی

ندارد. به علاوه، طبق (۳–۶۰) تنها سه مؤلفهٔ $h_{k-۲}, h_{k-۱}, h_k$ از ضرایب Bـاسپلاین $h(t)$ غیر صفر هستند. مقادیر عددی این ضرایب به ازای چند مرتبه در جدول ۴–۲ و نمودار توابع متناظر آن نیز در شکل ۴–۴ (ستون سمت راست) آمده‌اند.

نمودار بعدی یعنی شکل ۴–۵، رابطهٔ بین مرتبهٔ Bـاسپلاین مورد حذف گره قرار گرفته و تعداد کل محاسبات اصلی (flops) انجام شده در الگوریتم رمز متناظر با آنرا نمایش می‌دهد. در این نمودار نیز دو حالت، دنباله گره‌های یکنواخت و دنباله گره‌های متساوی‌الفاصله با تکرار $k-۱$، به عنوان کرانهایی برای تعداد عملیات اصلی در سایر شرایط در نظر گرفته شده‌اند. همانطور که قبلاً تخمین زده‌بودیم، الگوریتم رمز $\mathcal{O}(k^۴)$ است (البته اگر حلقهٔ اصلی الگوریتم را به طور ثابت k بار تکرار کنیم). برای نمایش این مطلب، در شکل ۴–۵ محور عرضی بر حسب $\sqrt[۴]{\text{flops}}$ مقیاس بندی شده‌است، و بدین ترتیب ملاحظه می‌کنید که ضریب ثابت مرتبهٔ این الگوریتم نیز چندان بزرگ نیست.

اگر به یاد بیاوریم، در فصل پیش در لم ۶.۳ ثابت شد که در هر دنبالهٔ گره یکنواخت، ضرایب براکتی متناظر با حذف هر گره با ضرایب بسط دو جمله‌ای برابرند:

$$\begin{bmatrix} k-۱ \\ j+۱ \end{bmatrix} = \binom{k}{j} \ , \qquad ۰ \le j \le k \ .$$

بدین ترتیب با توجه به (۳–۴۳) ، در چنین دنبالهٔ گره‌هایی داریم

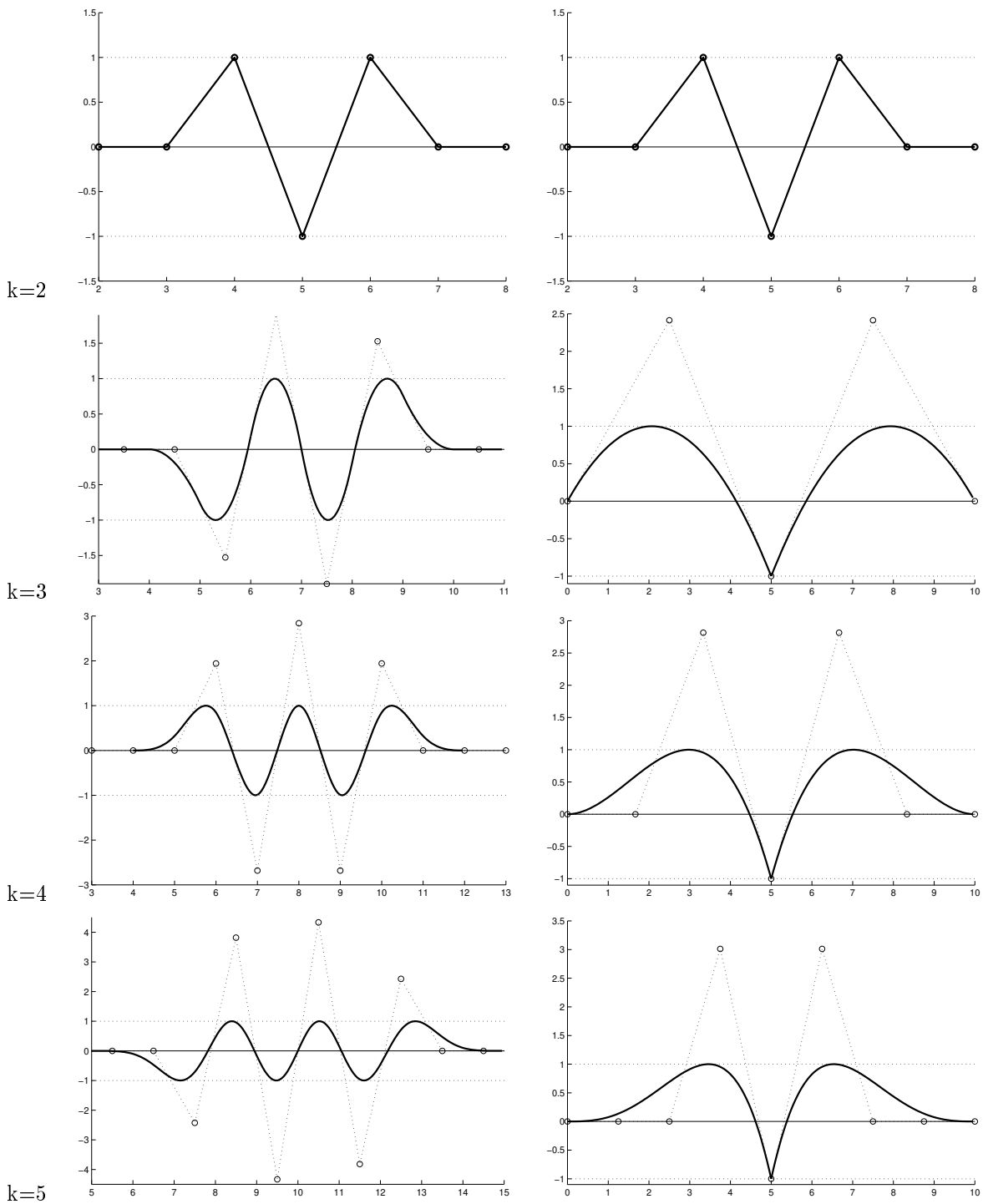
$$\widehat{\gamma}_\infty = \sum_{j=۰}^{k} \binom{k}{j} = ۲^k$$

زمانی که $m_۱ = m_۲ = ۰$ (گره حذف شده به قدر کافی از لبه‌های دامنه $\mathcal{I}$ دور است) و به وضوح $\tilde{\mu} = ۱$. بدین ترتیب، (۳–۷۳) به صورت زیر در می‌آید:

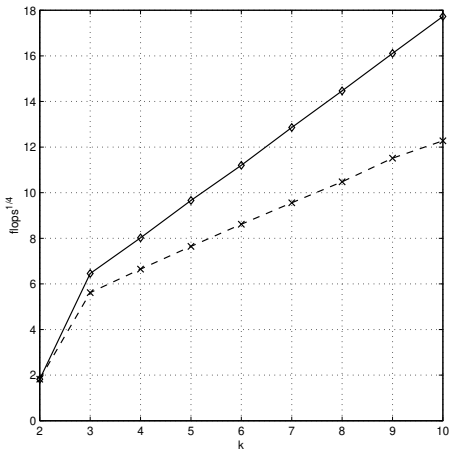
$$\operatorname{dist}(\tilde{Q}, \bar{Q}) \le \|\tilde{\mathbf{c}} - A\bar{\mathbf{c}}\|_\infty = ۲^{-k}\|D\|_۲ \ . \qquad (۲۰–۴)$$

حالت خاص دیگری که برای بررسی قابل توجه است، نقطهٔ مقابل دنبالهٔ گره یکنواخت، دنبالهٔ گره‌هایی است که تمام گره‌های آن دارای مرتبهٔ تکرار k و متساوی‌الفاصله هستند. با یک محاسبهٔ ساده می‌توان نشان داد که در چنین وضعیتی داریم

$$\tilde{\mu} = k \quad \Rightarrow \quad \begin{bmatrix} k-۱ \\ k \end{bmatrix} = \begin{bmatrix} k-۱ \\ k+۱ \end{bmatrix} = ۱$$



شکل ۴–۴: (سمت راست) توابع اسپلاین چبیشفی روی دنباله گره‌های متساوی‌الفاصله با مرتبه تکرار $k-1$، (سمت چپ) توابع اسپلاین چبیشفی روی دنباله گره‌های یکنواخت.



شکل ۴—۵: نمودار تعداد عملیات اصلی الگوریتم رمز برحسب مرتبهٔ Bـاسپلاین: بر روی دنباله گره‌های یکنواخت (منحنی توپر)، بر روی دنباله گره‌های متساوی‌الفاصله با تکرار $k-1$ (منحنی خط‌چین).

و در نتیجه با فرض $m_1 = m_2 = 0$، داریم

$$\widehat{\gamma}_\infty = \sum_{j=k-1}^{k} \binom{k}{j} = 2$$
$$\operatorname{dist}(\tilde{Q}, \bar{Q}) \le \|\tilde{\mathbf{c}} - A\bar{\mathbf{c}}\|_\infty = 2^{-1}\|D\|_2 .$$

می‌توان انتظار داشت که ضریب $\widehat{\gamma}_\infty$ برای سایر دنبالهٔ گره‌ها نیز به تقریب، بین دو مقدار فوق محدود باشد. بدین ترتیب در عمل به طور میانگین می‌توان گفت که $\operatorname{dist}(\tilde{Q}, \bar{Q}) \le 2^{-k/2}\|D\|_2$. اگر از حالاتی که تمام گره‌ها تقریباً اشباع شده هستند بگذریم، می‌توان گفت که با افزایش مرتبهٔ Bـاسپلاین k، تقریب L_∞ گسسته به طور دقیقتری خطای L_∞ پیوسته بین دو منحنی $\tilde{Q}$ و $\bar{Q}$ را تخمین می‌زند. برای بررسی دقیقتر این موضوع از یک قضیهٔ مهم و جالب استفاده می‌کنیم که رابطهٔ بین نرم برداری L_∞ و نرم تابعی L_∞ را در Bـاسپلاینها نشان می‌دهد.

قضیه‌ی ۲.۴ برای هر تابع Bـاسپلاین تعریف شده بر $\mathcal{S}^k_{n,\mathcal{U}}$ با ضرایب Bـاسپلاین $\{c_i\}_{i=1}^n$ به صورت $f(t) = \sum_{i=1}^n c_i B_i^k(t)$، داریم

$$\mathcal{D}_{k,\infty}^{-1}\|\mathbf{c}_L\|_\infty \le \|\sum_{i=1}^{n} c_i B_i^k(t)\|_\infty \le \|\mathbf{c}_L\|_\infty \qquad (۴-۲۱)$$

که در آن ضرایب ثابت $\mathcal{D}_{k,\infty}$ مستقل از دنبالهٔ گره‌ها و ضرایب Bـاسپلاین است و تنها به k وابسته است. مقدار $\mathcal{D}_{k,\infty}$ به ازای چند k به شرح زیر است:

k	۲	۳	۴	۵	۶	۷
$\mathcal{D}_{k,\infty}$	۱	۳	۵	$۱۱\frac{۲}{۳}$	۲۱	$۴۶\frac{۱}{۵}$

و به طور کلی داریم

$$\mathcal{D}_{k,\infty} \sim ۲^{k-\frac{۳}{۲}} . \qquad (۲۲-۴)$$

برهان. یک اثبات کامل و دقیق با تشریح مقادیر عددی $\mathcal{D}_{k,\infty}$ در [۱۲] آمده است. □

نتیجه‌ی ۳.۴ با بکارگیری قضیهٔ ۲.۴ بر روی $\mathrm{dist}(\tilde{Q},\bar{Q})$ با توجه به (۳-۳۰) ، داریم

$$\mathcal{D}_{k,\infty}^{-۱}\widehat{\gamma}_{\infty}^{-۱}\|D\|_{۲} \leq \mathrm{dist}(\tilde{Q},\bar{Q}) \leq \widehat{\gamma}_{\infty}^{-۱}\|D\|_{۲}$$

و برای وقتیکه $\tilde{Q}$ روی یک دنبالهٔ گره یکنواخت تعریف شده است، با توجه به (۴-۲۰) ، داریم

$$۲^{-۲k+\frac{۳}{۲}}\|D\|_{۲} \sim \mathcal{D}_{k,\infty}^{-۱}۲^{-k}\|D\|_{۲} \leq \mathrm{dist}(\tilde{Q},\bar{Q}) \leq ۲^{-k}\|D\|_{۲} . \qquad (۲۳-۴)$$

بدین ترتیب ملاحظه می‌شود که وقتی k خیلی کوچک نیست، حذف گره با تقریب L_{∞} گسسته به میزان قابل قبولی با حذف گره با بهترین تقریب L_{∞} پیوسته، مشابه عمل می‌کند. این استنتاج در مورد حذف گره $L_{۲}$ گسسته نیز تقریباً بر قرار است. نتایجی که در ادامه خواهند آمد، این موضوع را تأیید می‌کنند.

حال برای ارزیابی و مقایسه سه روش حذف گره (با تقریب L_{∞} گسسته، $L_{۲}$ گسسته و L_{∞} پیوسته) و سه روش انتخاب گره (با ارزیابی مجدد، یک بار مرتب‌سازی و مستقل) برای حذف، نتایج بدست آمده از اجراهای متفاوت الگوریتم تجزیه بر روی یک منحنی نمونه را بررسی می‌کنیم.

منحنی $\big(t+\cos(۴t)/۵, ۲t(۲-t)+\sin(۲۴t)/۵\big)$ را وقتیکه $t \in [۰\ \ ۲]$ ، بعنوان یک منحنی نمونه فرض کنید. با برخی روشها می‌توان به سادگی این منحنی پارامتری را به صورت یک منحنی Bـاسپلاین روی دنباله‌ای با تعداد بسیار زیادی گره، مدل سازی کرد (دو روش ساده برای این کار در بخش ۵-۲ در فصل کاربردها ذکر شده‌است). در اینجا یک دنبالهٔ گره‌های یکنواخت با ۸۰۰ گره برای مدل سازی اولیهٔ این منحنی بکار رفته است.

در جدول ۴—۳، اختلاف نهایی بین منحنی اولیه و منحنی حاصل از حذف ۷۸۵ گره از ۸۰۰ گره اولیه، به ازای بکارگیری هریک از تقریبها و هر یک از روشهای انتخاب گره، آمده است.

مرتبهٔ اسپلاین / نوع تقریب، روش انتخاب گره	k=3			k=4		
	L_∞ گسسته	L_2 گسسته	L_∞ پیوسته	L_∞ گسسته	L_2 گسسته	L_∞ پیوسته
ارزیابی مجدد	۰/۴۰۴۴	۰/۳۷۰۰	۰/۲۶۹۹	۱/۲۲۵۸	۰/۹۸۷۱	۰/۸۶۴۴
مرتب سازی	۰/۴۲۸۳	۰/۲۸۴۳	۰/۳۸۵۷	۰/۵۲۵۶	۰/۳۵۵۹	۰/۳۴۱۲
مستقل	۰/۵۲۹۴	۰/۴۹۷۱	۰/۵۷۳۶	۰/۳۹۰۹	۰/۳۰۰۹	۰/۳۸۰۶

جدول ۴—۳: مقایسه خطای کل بین منحنی اولیه و منحنی نهایی بعد از اجراهای متخلف الگوریتم تجزیه بر روی یک منحنی نمونه.

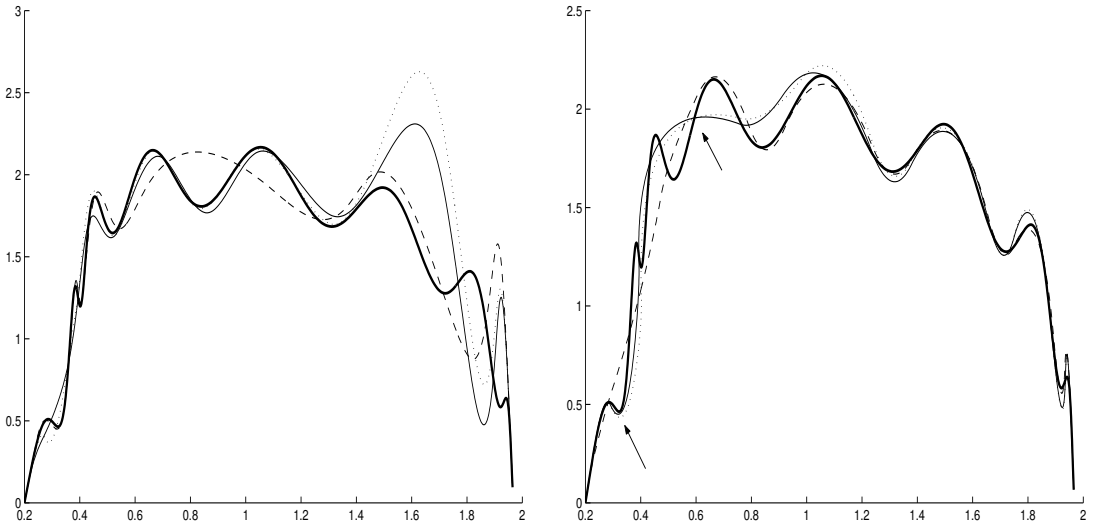
در هر یک از اشکال ۴—۶، نمودار منحنی اصلی همراه با سه منحنی حاصل از حذف گره‌ها بر اساس سه تقریب متفاوت نشان داده می‌شود.

با مقایسهٔ اشکال ۴—۶ و مقادیر خطای متناظرشان در جدول ۴—۳، برخی نتایج به شرح زیر جمعبندی می‌شوند.

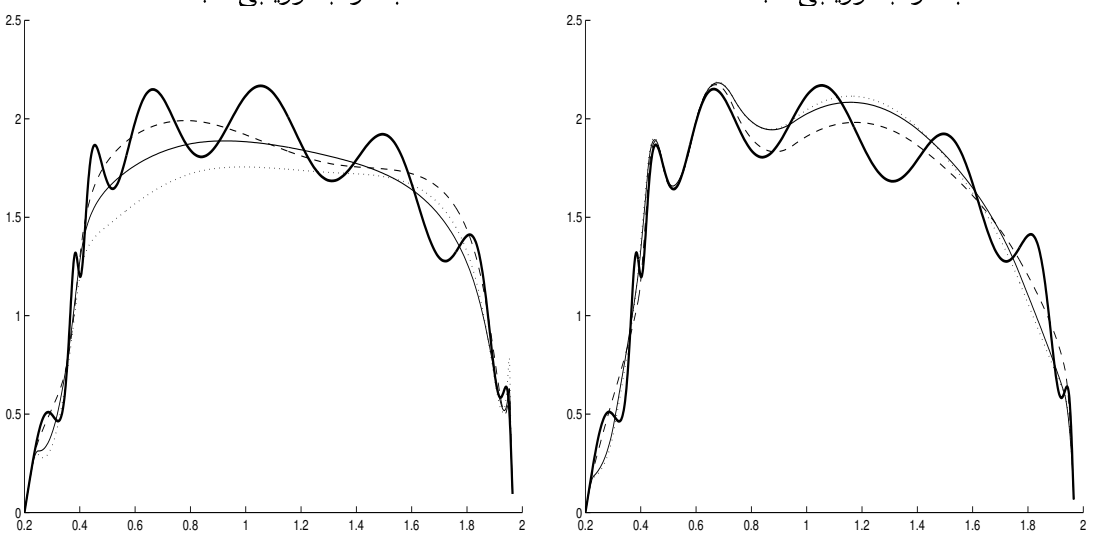
- خطای تجمعی حذفهای متوالی با تقریب L_∞ گسسته در بیشتر موارد بزرگتر از خطای کلی دو نرم دیگر است. به علاوه، از دیدگاه هندسی در برخی موارد رفتار منحنی حاصل از حذفهای با این تقریب، دارای نوسانات شدید است و ناپایدار به نظر می‌رسد.

- بر خلاف تقریب L_∞ گسسته، تقریب L_2 گسسته از دیدگاه هندسی خوشرفتارتر و حتیٰ از L_∞ پیوسته کم نوسانتر به نظر می‌رسد. به علاوه، با مقایسه مقادیر آمده در جدول در می‌یابیم که حذفهای متوالی بر اساس این نرم تقریباً مستقل از نوع انتخاب گره است.

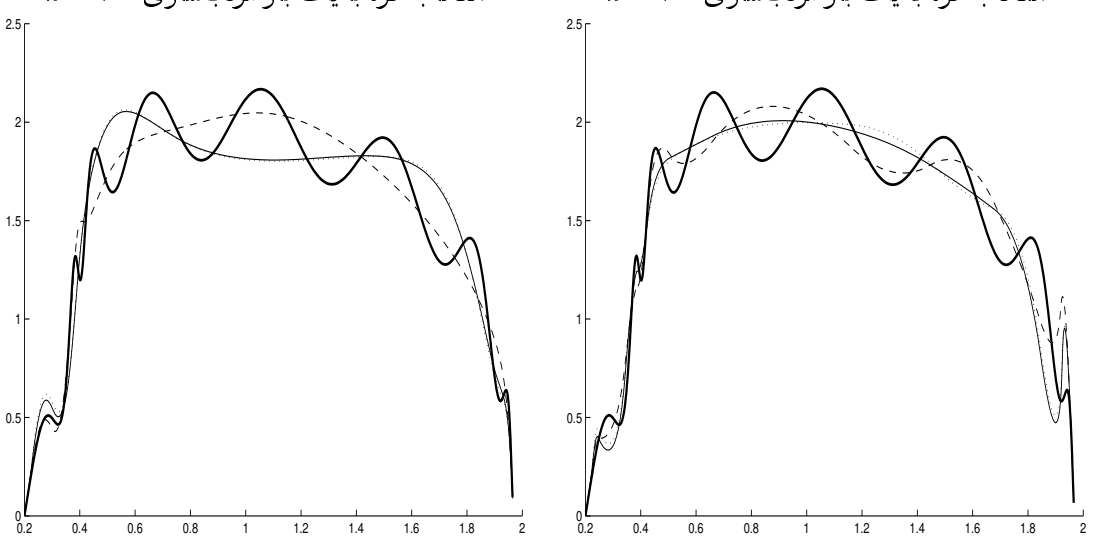
- خطای تجمعی حذف گره با تقریب L_∞ پیوسته در بیشتر (نه همه) موارد کمترین مقدار خطا در بین دیگر نرمهاست، که البته چندان دور از انتظار نیست. با این حال در مقایسه با L_2 گسسته، به نوع انتخاب گره حساس است به طوری که با روش انتخاب گره با ارزیابی مجدد، کمترین خطا را می‌دهد ولی همراه با دو روش دیگر بخصوص روش مستقل، به نظر می‌رسد که از بهترین جواب کلی دور می‌شود.



انتخاب گره با ارزیابی مجدد $k = ۳$ انتخاب گره با ارزیابی مجدد $k = ۴$



انتخاب گره با یک بار مرتب‌سازی $k = ۳$ انتخاب گره با یک بار مرتب‌سازی $k = ۴$



انتخاب گره‌های متوالی مستقل $k = ۳$ انتخاب گره‌های متوالی مستقل $k = ۴$

شکل ۴—٦: حذف ۷۸۵ گره از ۸۰۰ گره در یک منحنی با استفاده از روشهای متفاوت برای انتخاب گره و نوع تقریب: در تمام این اشکال منحنی اولیه با (خط توپر)، منحنی حاصل از حذفها با تقریب L_∞ گسسته با (نقطه‌چین)، $L_۲$ گسسته با (خط‌چین) و L_∞ پیوسته با (خط نازک) نشان داده شده‌است.

- با اینکه انتظار می‌رود خطای حذف گره در Bـاسپلاینهای مرتبه بالاتر، کمتر باشد، امّا این مثال خاص نشان می‌دهد که این موضوع عمومیت ندارد. در این مورد، یک دلیل می‌تواند این باشد که تصمیم‌گیریهای موضعی در الگوریتم تجزیه دربارهٔ انتخاب بهترین گره قابل حذف، برای Bـاسپلاین مرتبه بالاتر که از درجه آزادی بیشتری برخوردار است، در بعضی مواقع منجر به انحراف بزرگتری در مسیر رسیدن به بهترین جواب کلی می‌شود. با این حال از دیدگاه هندسی، رفتار منحنیهای مرتبهٔ ۳ حاصل از الگوریتم تجزیه، برای مدل سازی منحنی اولیه مناسبتر به نظر می‌رسد.

فصل ۵

کاربردها

۵–۱ رتبه بندی گره‌ها برای الگوریتمهای حذف همزمان چند گره

یکی از اساسی‌ترین کاربردهای حذف تک گره، حل مسألهٔ رتبه‌بندی گره‌ها به عنوان یک زیربرنامهٔ کلیدی در روشهای حذف همزمان چند گره است. همانطور که در فصل پیش اشاره شد، روشهای حذف چند گره از جهات متعددی و به خصوص به واسطهٔ سادگی، کارایی و سریع بودنشان در عمل از الگوریتمهای پر کاربرد به شمار می‌روند.

ایدهٔ اولیهٔ مسألهٔ حذف گره ابتدا به طور اساسی توسط **لیخ و مورکن** در یک سری مقالات [۲۷]، [۲۶] و [۲۸] به طور عملی مورد توجه قرار گرفت. به بیان ساده، در الگوریتمهای حذف چند گره به نوعی سعی می‌شود، که الگوریتم درج چند گره (الگوریتم اسلو) معکوس‌سازی شود. مشابه بحث حذف تک گره به وضوح این عمل به طور یکتا انجام‌پذیر نیست.

به طور دقیقتر، فرض کنید منحنی Bـاسپلاین $\tilde{Q} = \tilde{\mathbf{B}}_L^T \tilde{\mathbf{c}}_L$ بر روی دنبالهٔ گره‌های $\tilde{U}$ داده شده است. با فرض اینکه l گره $t_1, \ldots, t_l$ به طور مناسب انتخاب شده‌اند، می‌خواهیم به طور همزمان تمام گره‌های $t_1, \ldots, t_l$ را به گونه‌ای حذف کنیم که منحنی Bـاسپلاین $\bar{Q} = \bar{\mathbf{B}}_L^T \bar{\mathbf{c}}_L$ بر روی دنبالهٔ گره‌های $\bar{U} = \tilde{U} \setminus \{t_1, \ldots, t_l\}$ دارای کمترین اختلاف با منحنی $\tilde{Q}$ در برخی نرمها باشد. نرم مورد استفاده در این

الگوریتمها به صورت

$$\|\tilde{Q}(t)\|_{l^p,\tilde{\mathcal{U}}} = \begin{cases} \left(\sum_j |\tilde{c}_j|^p(\tilde{u}_{j+k}-\tilde{u}_j)/k\right)^{1/p}, & 1 \le p \le \infty \\ \max_j |\tilde{c}_j|, & p=\infty \end{cases}$$

تعریف می‌شود و نرم $l^p,\tilde{\mathcal{U}}$ نامیده می‌شود. همانطور که ملاحظه می‌کنید، نرم $l^\infty,\tilde{\mathcal{U}}$ یک تابع Bـاسپلاین با نرم برداری L_∞ برای بردار $\tilde{\mathbf{c}}_L$ برابر است. مشابه قضیهٔ ۲.۴ و به عنوان تعمیمی از آن، رابطهٔ زیر برای این نرم قابل اثبات است:

$$\mathcal{D}_k^{-1}\|\tilde{Q}\|_{l^p,\tilde{\mathcal{U}}} \le \|\tilde{Q}\|_{L^p} \le \|\tilde{Q}\|_{l^p,\tilde{\mathcal{U}}}$$

که در آن نرم تابعی $\|.\|_{L^p}$ به طور معمول تعریف می‌شود. در اینجا نیز $\mathcal{D}_k$ عدد ثابتی است که تنها به k وابسته است.

همانطور که پیشتر اشاره شد حل مسائل حذف چند گره بر اساس مینیمم‌سازی $\|\tilde{Q}-\bar{Q}\|_{L^p}$ چندان عملی نیست، در عوض می‌توان مسألهٔ مینییم‌سازی $\|nQ-\bar{Q}\|_{l^p,\tilde{\mathcal{U}}}$ را با برخی ملاحظات حل نمود. آنچنانکه معمول است، بیشتر به حالت $p=2$ یا $p=\infty$ توجه می‌شود. مشابه الگوریتم حذف تک گره، در اینجا نیز الگوریتم به دو مرحلهٔ اساسی تقسیم می‌شود. در مرحلهٔ اول فرض می‌کنیم گره‌های $t_1,\ldots,t_l$ به طریقی در دسترس هستند. با حل مسألهٔ کمترین مربعات زیر، $\bar{Q}$ را به عنوان جواب مسألهٔ حذف گره‌های $t_1,\ldots,t_l$ با کمترین خطا در نرم $l^2,\tilde{\mathcal{U}}$ نسبت به $\tilde{Q}$ بدست می آوریم:

$$\min_{\bar{Q}\in\mathcal{S}^k_{n,\mathcal{U}}} \|\tilde{Q}-\bar{Q}\|_{l^2,\tilde{\mathcal{U}}} \quad \Longleftrightarrow \quad \min_{\bar{\mathbf{c}}_L\in\mathbb{R}^n} \|E_{1/2,\tilde{\mathcal{U}}}(A^{[k]}\bar{\mathbf{c}}_L-\tilde{\mathbf{c}}_L)\|_2$$

که $E_{1/2,\tilde{\mathcal{U}}}$ ماتریس قطری با مؤلفه‌هایی به صورت $e_{i,i}=\left(\frac{1}{2}(\tilde{u}_{i+k}-\tilde{u}_i)\right)^{1/2}$ و $A^{[k]}$ ماتریس درج گره در الگوریتم اسلو است (به (۲–۲۳) در بخش ۲–۳ نگاه کنید).

در مرحلهٔ دوم از الگوریتم حذف همزمان چند گره، به پاسخ این پرسش پرداخته می‌شود که حداکثر چه تعداد از گره‌ها را می‌توان حذف کرد به گونه‌ای که خطا از حد معینی تجاوز نکند. برای حل این مسأله عموماً از یک کنکاش دودویی[1] برای تخمین حداکثر تعداد گره‌های قابل حذف استفاده می‌شود. بدین ترتیب که با در دست داشتن یک لیست مرتب شده از گره‌ها $t_1,\ldots,t_l$ به گونه‌ای که خطای حذف تک گره t_i کمتر از خطای حذف تک گره t_{i+1} است، ابتدا بوسیلهٔ روابط حذف چند گره در مرحلهٔ اول، تمام $t_1,\ldots,t_l$ گره را

[1] Binary search

حذف می‌کنیم. اگر خطای کل از ε کوچکتر بود که به جواب مطلوب رسیده‌ایم در غیر اینصورت نیمی از گره‌ها $t_1, \ldots, t_{\lceil l/2 \rceil}$ را حذف می‌کنیم. اگر مقدار خطا در این هنگام کمتر از ε بود، همین عملیات را با ۳/۴ گره‌ها $t_1, \ldots, t_{\lceil \frac{3}{4} l \rceil}$ انجام می‌دهیم در غیر اینصورت همین عملیات را با ۱/۴ گره‌ها $t_1, \ldots, t_{\lceil \frac{1}{4} l \rceil}$ انجام می‌دهیم. این عمل تا زمانیکه دقت مطلوب حاصل شود، تکرار می‌شود.

مسأله رتبه‌بندی[۲] گره‌ها در الگوریتم حذف چند گره، تعیین یک معیار مناسب برای مرتب‌سازی گره‌های مورد حذف پیش از مرحلهٔ کنکاش دودویی است. به عبارت دیگر، می‌خواهیم لیست مرتب شدهٔ $t_1, \ldots, t_l$ را بگونه‌ای تعیین کنیم که تقریباً خطای حذف هر تک گره t_i در این لیست کمتر از خطای حذف گره‌های بعدی t_j که $j > i$ ، باشد. روش انتخاب گره با یک بار مرتب‌سازی که در بخش ۴—۲ معرفی شد، یک راه حل مناسب برای این مسأله است.

آنچنانکه در روش انتخاب گره با یک بار مرتب‌سازی تشریح شد، می‌توانیم بزرگی خطای حذف هر گره را با اندازهٔ ضریب خطای حذف گره $\|D\|_2$ و یا خطای حذف گره L_∞ گسسته $\|D\|_2 \widehat{\gamma}_\infty^{-1}$ تقریب بزنیم. با این حال در این کاربرد خاص شاید محاسبهٔ دقیق خطای حذف تک گره $\|D\|_2 \gamma_\infty^{-1}$ به وسیلهٔ بکارگیری الگوریتم رمز، ارزش صرف محاسبات بیشتر را داشته باشد. به خصوص در مواردی که $\tilde{\mathcal{U}}$ یک دنبالهٔ یکنواخت از گره‌هاست، می‌توان به طور صریح از ضرایب اسپلاین چبیشفی که یک بار برای همیشه محاسبه شده‌اند در جدول ۴—۱، استفاده کرد.

سایر کاربردهای الگوریتمهای حذف تک گره که با بکارگیری روشهای ترکیب و تجزیه پیاده‌سازی شده در فصل پیش انجام می‌شوند، عمدتاً به عنوان کاربردهای مشترک تمام روشهای حذف گره شناخته می‌شوند. در ادامه به طور اجمال از میان تعداد انبوهی از اینگونه کاربردها، چند کاربرد کلی را بر اساس روشهای حذف تک گره بررسی می‌کنیم و به برخی دیگر به اختصار اشاره می‌کنیم. برای مطالعهٔ فهرست کاملتری از این کاربردها می‌توانید به [۲۸] ، [۲۵] و [۳۸] مراجعه کنید، هرچند ایدهٔ برخی از کاربردهایی که در ادامه ذکر می‌شوند برگرفته از کاربردهای روشهای ترکیب و تجزیهٔ موجک در [۴۴] است.

---

[۲] Ranking

۵–۲ برازش و کاهش داده‌ها

۵–۲–۱ برازش منحنی

یکی از متداولترین کاربردها در نرم‌افزارهای طراحی به کمک کامپیوتر، تبدیل خودکار اطلاعات خام به مدلهای هندسی است. به طور کلی در این کاربردها داده‌های اولیه از طریق یک دستگاه ورودی مثل اسکنر، دوربین فیلمبرداری و حتیٰ صفحه کلید و ماوس، یا به عنوان خروجی یک برنامهٔ دیگر به کامپیوتر وارد می‌شود. این داده‌ها می‌تواند مربوط به نقشه‌ها، عکسها، فونتهای خطاطی شده و نیز طرح اولیه [۳] های یک هنرمند باشد. دو ویژگی عمدهٔ این داده‌ها، بزرگی حجم اطلاعات و همراه داشتن داده‌های اغتشاشی است. هدف از برازش[۴]، تقریب زدن داده‌های اولیه با کاستن از تعداد داده‌ها و بدست آوردن یک مدل هموار شده است. این گونه کاربردها را که با کمی مسامحه کاهش داده‌ها[۵] می‌نامیم، می‌توان به ازای دقت تقریب معین یا تعداد داده‌های نهایی معین و نیز برقراری برخی قیود اجرا کرد.

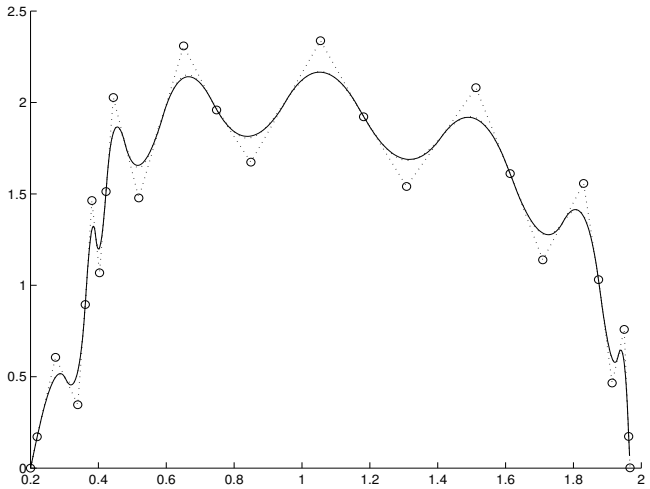
به طور کلی فرض کنید که می‌خواهیم n نقطهٔ $\{c_i\}_{i=1}^{n}$ را به وسیلهٔ یک منحنی Bـاسپلاین مرتبهٔ k برازش کنیم. برای این کار، ابتدا این مجموعه از نقاط را با انتخاب یک دنبالهٔ مناسب از گره‌ها، به وسیلهٔ یک منحنی Bـاسپلاین مدل سازی می‌کنیم. این مدلسازی اولیه لازم نیست خیلی فشرده باشد، بلکه قاعدتاً می‌تواند شامل تعداد زیادی رئوس کنترلی باشد. در عوض برخی از قیود هندسی روی منحنی را می‌توان در این مدل اولیه منظور کرد. روشهای زیادی از ساده تا پیچیده برای این تقریب اولیه وجود دارند. یک روش قوی که شامل پارامترهایی برای کنترل میزان هموارى و انحنای مدل Bـاسپلاین است در [۴۰] و [۴۳] ارائه شده است.

در اینجا یک روش ساده برای ساختن این منحنی اولیه تشریح می‌شود. با انتخاب یک دنبالهٔ متساوی‌الفاصله از گره‌ها که گره‌های اول و آخر آن با مرتبه تکرار k و گره‌های درونی آن با مرتبه تکرار $k-1$ است، و قرار دادن نقاط $\{c_i\}_{i=1}^{n}$ هر یک با تکرار $k-1$ به عنوان رئوس کنترلی، یک منحنی

[۳] Sketch
[۴] Fitting
[۵] Data reduction

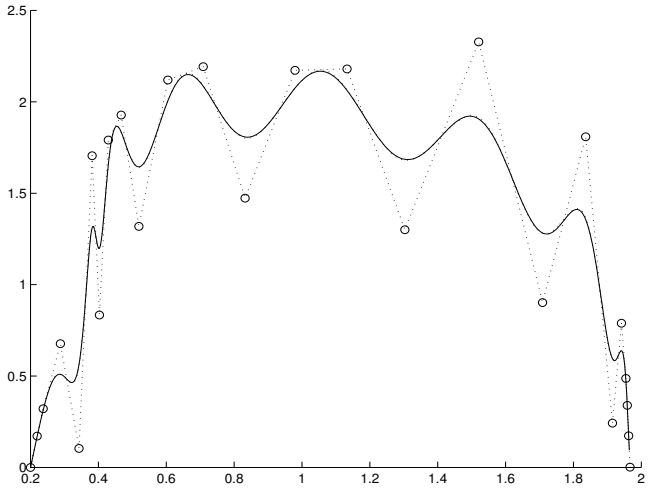
Bـاسپلاین C° بدست می آید که تمام رئوس کنترلی را درونیابی می‌کند. حتیٰ در مواردی که تعداد داده‌های اولیه بسیار زیاد باشد، به طور ساده‌تر، می‌توان یک دنبالهٔ گره یکنواخت انتخاب کرد و از خود نقاط داده به طور مستقیم به عنوان رئوس کنترلی استفاده کرد. منحنی اولیه در این روش C^{k-1} است و یک هموارسازی ساده نیز روی داده‌ها انجام می‌دهد، در حالیکه منحنی اولیه ساخته شده بر اساس روش قبل، ناپیوستگیهای مشتق و انحناء در داده‌های اولیه را حفظ می‌کند.

به هر حال بعد از بدست آوردن یک منحنی Bـاسپلاین اولیه می‌توان هر یک از روشهای تجزیه مذکور در بخش ۴—۲ را به کار گرفت و فرایند حذف گره را تا زمانیکه خطای حذف گره بین دو تکرار متوالی الگوریتم از آستانهٔ قابل قبول ε نگذشته است، ادامه داد. بدین ترتیب تعدادی از گره‌ها حذف می‌شوند و تعداد داده‌ها کاهش می‌یابد. به علاوه، بدلیلی که قبلاً شرح داده شد، نتیجهٔ نهایی یک تقریب نزدیک و هموارشده، از داده‌های اولیه خواهد بود. البته در برخی کاربردها هدف، کاهش داده‌ها تا رسیدن تعداد آنها به مقدار معینی مثل N است، که در این مورد بدون توجه به ε به طور ثابت $n-N$ بار الگوریتم تجزیه را تکرار می‌کنیم.



شکل ۵—۱: برازش منحنی با Bـاسپلاین مرتبه $k=3$: داده‌های اولیه با ۸۰۰ نقطه مدل شده‌اند (منحنی توپر)، منحنی برازنده با ۲۵ نقطهٔ کنترلی (منحنی نقطه‌چین؛ تفاوت به سختی دیده می‌شود)، چندضلعی کنترلی نهایی (نقطه‌چین).

در شکل ۵—۱ یک منحنی Bـاسپلاین مربعی اولیه با روش سادهٔ دوم، بر روی ۸۰۰ نقطهٔ داده ساخته می‌شود و پس از حذف ۷۷۵ گره، یک منحنی Bـاسپلاین برازنده با ۲۵ نقطهٔ کنترلی بدست می آید. در شکل ۵—۲ همین عمل به وسیلهٔ یک منحنی Bـاسپلاین مکعبی صورت گرفته است. در هر دو



شکل ۵—۲: برازش منحنی با Bـاسپلاین مرتبه $k = ۴$.

شکل، از روش انتخاب گره با یک بار مرتب‌سازی و حذف گره با بهترین تقریب L_∞ پیوسته استفاده شده است. خطای کل بین منحنی اولیه و نهایی در شکل ۵—۱ برابر $\text{dist} = ۰/۰۱۵۱$ و در شکل ۵—۲ برابر $\text{dist} = ۰/۰۰۷۴$ است. بدین ترتیب در مقام مقایسه می‌توان گفت که منحنی Bـاسپلاین مرتبهٔ $k = ۴$ با دقت بیشتری نسبت به $k = ۳$، داده‌ها را برازش کرده است.

برخی کاربردهای خاص برازش منحنی در نرم‌افزارهای طراحی هندسی عبارتند از:

۱) *برازش منحنی مقطع*. در تعداد بسیار زیادی از مسائل فیزیک و مهندسی، مدل سازی یک منحنی ایجاد شده از محل تقاطع دو رویه دلخواه از اهمیت زیادی برخوردار است. در اینگونه مسائل، مجموعه نقاط مقطع به صورت یک منحنی تراز[۶] جمع آوری می‌شوند و سپس با بکارگیری الگوریتم کاهش داده‌ها، این منحنی را با تعداد کمتری از داده‌ها برازش می‌کنند.

۲) *برازش منحنی آفست*[۷]. برای منحنی مسطح $Q(t) = \big(Q_x(t), Q_y(t)\big)$، منحنی آفست $P(t) = \big(P_x(t), P_y(t)\big)$ با یک فاصلهٔ قائم ثابت به اندازهٔ a از منحنی $Q(t)$ به صورت زیر تعریف می‌شود:

$$P_x(t) = Q_x(t) + \frac{Q'_y(t)}{\big(Q'^{۲}_x(t) + Q'^{۲}_y(t)\big)^{۱/۲}} a \quad , \quad P_y(t) = Q_y(t) - \frac{Q'_x(t)}{\big(Q'^{۲}_x(t) + Q'^{۲}_y(t)\big)^{۱/۲}} a \qquad (۱—۵)$$

[۶] Contour curve

[۷] Offset curve

که $Q'_x(t)$ و $Q'_y(t)$ به ترتیب نشاندهندهٔ مشتق $Q_x(t)$ و $Q_y(t)$ نسبت به t است.

عموماً برای بدست آوردن این منحنی نمی‌توان روابط صریحی بر حسب توابع شناخته‌شده بدست آورد. یک رهیافت برای حل این مسأله، برازش منحنی به وسیلهٔ Bـاسپلاینهاست. بدین ترتیب که بعد از ارزیابی رابطهٔ(۵–۱) به ازای تعداد زیادی از مقادیر پارامتر، نقاط بدست آمده را بوسیلهٔ روش مذکور برازش می‌کنیم.

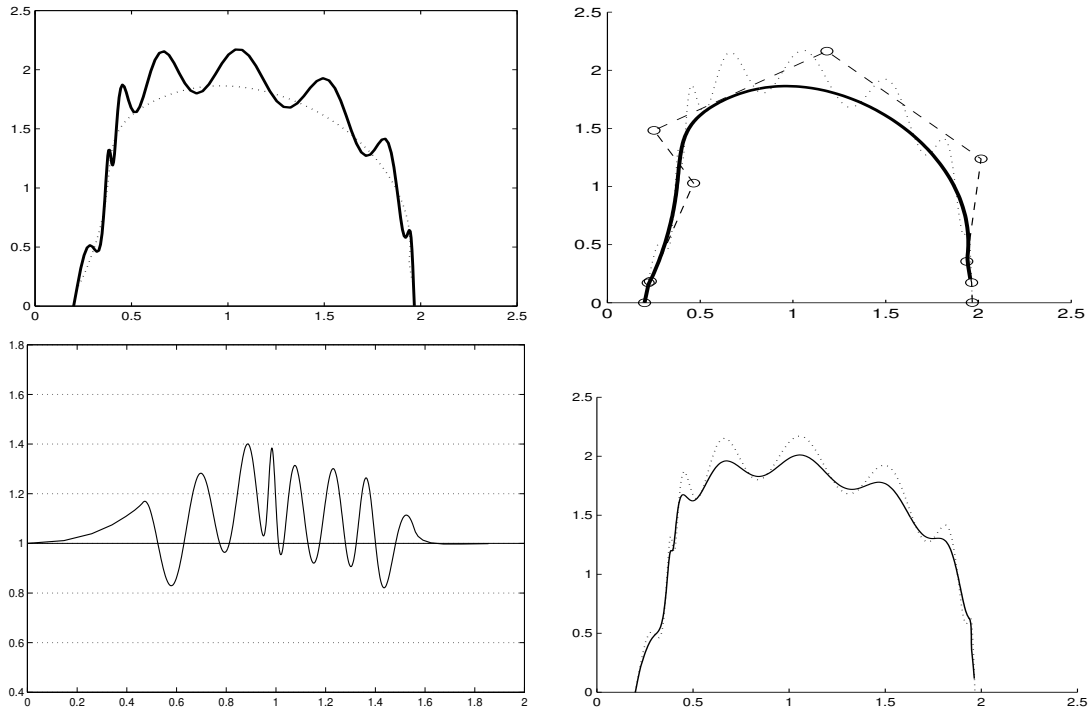
۵–۲–۲ پردازش امواج

یکی از مزیتهای روشهای ترکیب و تجزیه پیاده‌سازی شده در این رساله، نسبت به سایر روشهای حذف گره در اینست که در این روشها همانند روشهای تجزیه موجک، با هر بار عمل تجزیه، هم داده‌ها را کاهش می‌دهیم و هم به طریقی اغتشاشات درون داده‌های خام را استخراج می‌کنیم. بدین ترتیب برخی کاربردهای مشابه تبدیلات فیلتر در پردازش امواج[۸] عملی می‌شوند. این کاربردها با اعمال تغییرات روی داده‌های تقریب‌زننده و داده‌های خطا و ترکیب مجدد آنها بدست می‌آیند.

در یک کاربرد می‌توان پس از تجزیه داده‌ها، داده‌های خطای حاصل از تجزیه را کاملاً به دور ریخت و داده‌های تقریب‌زننده را با خطای صفر ترکیب نمود. نتیجه این کار یک اطلاعات بازسازی شده است که از حذف اغتشاشات داده‌های اولیه بدست آمده است. برعکس، در کاربرد دیگر می‌توان داده‌های خطا را نگهداشت و با ترکیب مضربی از آنها با یک مدل دیگر یا همان داده‌ها، اغتشاشات داده‌های اولیه را روی یک مدل دیگر سوار و یا تقویت کرد.

در شکل ۵–۳ یک منحنی موجی و تجزیهٔ آن به منحنی تقریب‌زننده و دادهای اغتشاشی نشان داده شده است. در این تجزیه ۷۹۰ گره، با یک بار مرتب‌سازی ۸۰۰ گره اولیه، بروش بهترین تقریب L_∞ پیوسته حذف شده‌اند. با نصف کردن ارتفاع داده‌های اغتشاشی و ترکیب مجدد آنها با منحنی تقریب‌زننده، یک موج بازسازی شده با اغتشاشات کوچکتر بدست می‌آید که در گوشهٔ راست و پائین شکل ۵–۳ نشان داده شده است. اگر همین اغتشاشات را بدون تغییر، با یک دایره ترکیب کنیم، منحنی شکل ۵–۴ بدست می‌آید.

[۸] Signal processing



شکل ۵—۳: موج اصلی (منحنی توپر)، موج تقریب (منحنی نقطه‌چین) در شکل سمت چپ—بالا؛ موج اصلی (منحنی نقطه‌چین)، موج تقریب (منحنی توپر) و چند ضلعی کنترلی آن (خط‌چین) در شکل سمت راست—بالا؛ منحنی اغتشاش در شکل سمت چپ—پائین؛ موج اصلی (منحنی نقطه‌چین)، موج بازسازی شده از نصف کردن بلندی اغتشاش (منحنی توپر) در شکل سمت راست—پائین.

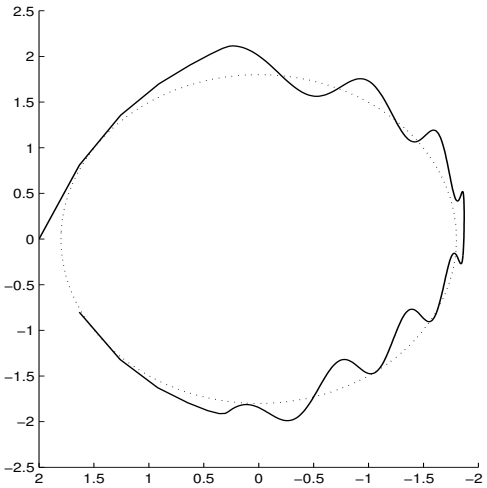
۵—۲—۳ فشرده‌سازی و پردازش تصاویر

مشابه اعمال مذکور را می‌توان بر روی رویه Bـاسپلاین ضرب تانسوری نیز انجام داد. یک رویهٔ Bـاسپلاین ضرب تانسوری $S : [u_k \; u_{n+1}) \times [v_l \; v_{m+1}) \mapsto \mathbb{R}^3$ به صورت زیر تعریف می‌شود:

$$S(t,s) = \sum_{i=1}^{n} \sum_{j=1}^{m} c_{i,j} B_i^k(t) B_j^l(s) \qquad t \in [u_k \; u_{n+1}) \; , \; s \in [v_l \; v_{m+1})$$

که آرایهٔ $C = [c_{i,j}]$ با ابعاد $n \times m$ شامل مؤلفه‌های $c_{i,j} \in \mathbb{R}^3$ است و تور کنترلی[۹] رویهٔ S نامیده می‌شود. توابع پایهٔ Bـاسپلاین $B_i^k(t)$ و $B_j^l(s)$ به ترتیب از مرتبه‌های k و l و روی دنبالهٔ گره‌های $\mathcal{U}$ و $\mathcal{V}$ تعریف شده‌اند. آنچه در مورد رویه‌های ضرب تانسوری قابل توجه است، شباهت ساختاری آنها با منحنیها است.

[۹] Control net



شکل ۵—۴: ترکیب اغتشاش با یک دایره.

با توجه به تعریف فوق، به یک تعبیر می‌توان گفت که یک رویهٔ Bـاسپلاین ضرب تانسوری، یک منحنی Bـاسپلاین فضایی است که رئوس کنترلی آن بر روی یک منحنی Bـاسپلاین دیگر حرکت می‌کند. نمایش ماتریسی این رویه این موضوع را بهتر نشان می‌دهد:

$$S = \begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix} \quad , \qquad S_x = \mathbf{B}^{k^T} C_x \mathbf{B}^l \quad S_y = \mathbf{B}^{k^T} C_y \mathbf{B}^l \quad S_z = \mathbf{B}^{k^T} C_z \mathbf{B}^l$$

که S_x, S_y, S_z مؤلفه‌های رویهٔ S و ماتریسهای C_x و C_y و C_z به ترتیب شامل مؤلفه‌های x و y و z هر یک از عناصر آرایهٔ C هستند.

منحنی عمودی $S_x^t(s) \in I\!R^n$ را به صورت

$$S_x^t(s) = \sum_{j=1}^{m} C_x^j B_j^l(s)$$

تعریف می‌کنیم که در آن بردار $C_x^j \in R^n$ حاوی عناصر ستون j ام ماتریس C_x است. به همین ترتیب، منحنی افقی $S_{x,s}(t) \in I\!R^m$ را به صورت

$$S_{x,s}(t) = \sum_{i=1}^{n} C_{x,i} B_i^k(t)$$

تعریف می‌کنیم که در آن بردار $C_{x,i} \in R^m$ حاوی عناصر سطر i ام ماتریس C_x است[۱۰]. به طور مشابه، منحنیهای عمودی $S_y^t(s), S_z^t(s) \in I\!R^n$ و افقی $S_{y,s}(t), S_{z,s}(t) \in I\!R^m$ را تعریف می‌کنیم.

[۱۰] دقت کنید که این تعریف برای منحنیهای عمودی و افقی با برخی تعاریف متداول فرق دارد.

حال برای بکارگیری حذف گره و الگوریتم تجزیه بر روی رویه‌های Bـاسپلاین ضرب تانسوری، متناوباً می‌توان الگوریتم تجزیه را یک یا چند بار به طور همزمان بر روی سه منحنی عمودی $S_x^t(s), S_y^t(s), S_z^t(s)$ و سپس یک یا چند بار به طور همزمان بر روی سه منحنی افقی $S_{x,s}(t), S_{y,s}(t), S_{z,s}(t)$ اعمال کرد، و این فرایند را تا رسیدن به شرایط مطلوب تکرار کرد.

یک تصویر را می‌توان به عنوان یک آرایهٔ دوبعدی از نقاط رنگی در نظر گرفت، به طوری که هر مؤلفهٔ آن حاوی عدد متناظر با رنگ هر نقطه می‌باشد. در تصاویر سیاه‌سفید، به هر رنگ، متناظر با مکان قرارگرفتن آن در طیف خاکستری بین سیاه تا سفید، یک عدد بین صفر تا یک نسبت داده می‌شود. در تصاویر رنگی، با تفکیک هر رنگ به رنگهای اصلی تشکیل دهندهٔ آن (قرمز، سبز و آبی)، به هر نقطهٔ تصویر سه عدد متناظر با میزان شدت هر یک از رنگهای اصلی در آن نقطه، نسبت داده می‌شود.

از دیدگاه طراحی هندسی، یک تصویر در اصل یک تابع $p(i,j) \mapsto \mathbb{R}$ (تصویر سیاه‌سفید) یا $p(i,j) = \big(p_r(i,j), p_g(i,j), p_b(i,j)\big) \mapsto \mathbb{R}^3$ (تصویر رنگی) است. $p(i,j)$ نشاندهندهٔ عدد متناظر با رنگ خاکستری در نقطهٔ (i,j) تصویر است و $p_r(i,j), p_g(i,j), p_b(i,j)$ به ترتیب متناظر با رنگهای آبی، سبز و قرمز در نقطهٔ (i,j) است. بدین ترتیب می‌توان از روشهای حذف گره برای رویه‌های Bـاسپلاین ضرب تانسوری، برای برخی کاربردها در فشرده‌سازی و پردازش تصاویر نیز استفاده کرد. برای این کار نیز، ابتدا یک رویهٔ Bـاسپلاین ضرب تانسوری را برای مدل سازی اولیهٔ داده‌های تصاویر انتخاب می‌کنیم. این کار را می‌توان با انتخاب دنبالهٔ گره‌ها و نقاط کنترلی مطابق روش اول پیشنهاد شده در برازش منحنی برای هر یک از منحنیهای عمودی و افقی رویه انجام داد.

با در دست داشتن این رویهٔ اولیه و با بکارگیری روشهای تجزیه بر روی آن می‌توان داده‌های تصویر را کاهش داد و ضرایب خطای آن را جدا کرد. تنها نکته‌ای که تذکر آن لازم به نظر می‌رسد اینست که برای بکارگیری روش تجزیه بر روی سه رویهٔ قرمز، سبز و آبی یک تصویر رنگی، دنبالهٔ گره‌های حذف شده از هر سه‌تای این رویه‌ها باید دقیقاً با یکدیگر برابر باشند، والا در صورت تجزیه مستقل هر یک از این سه رویه، نتیجهٔ نهایی از دیدگاه مفاهیم گرافیک کامپیوتری به عنوان یک تصویر رنگی قابل تعبیر نمی‌باشد.

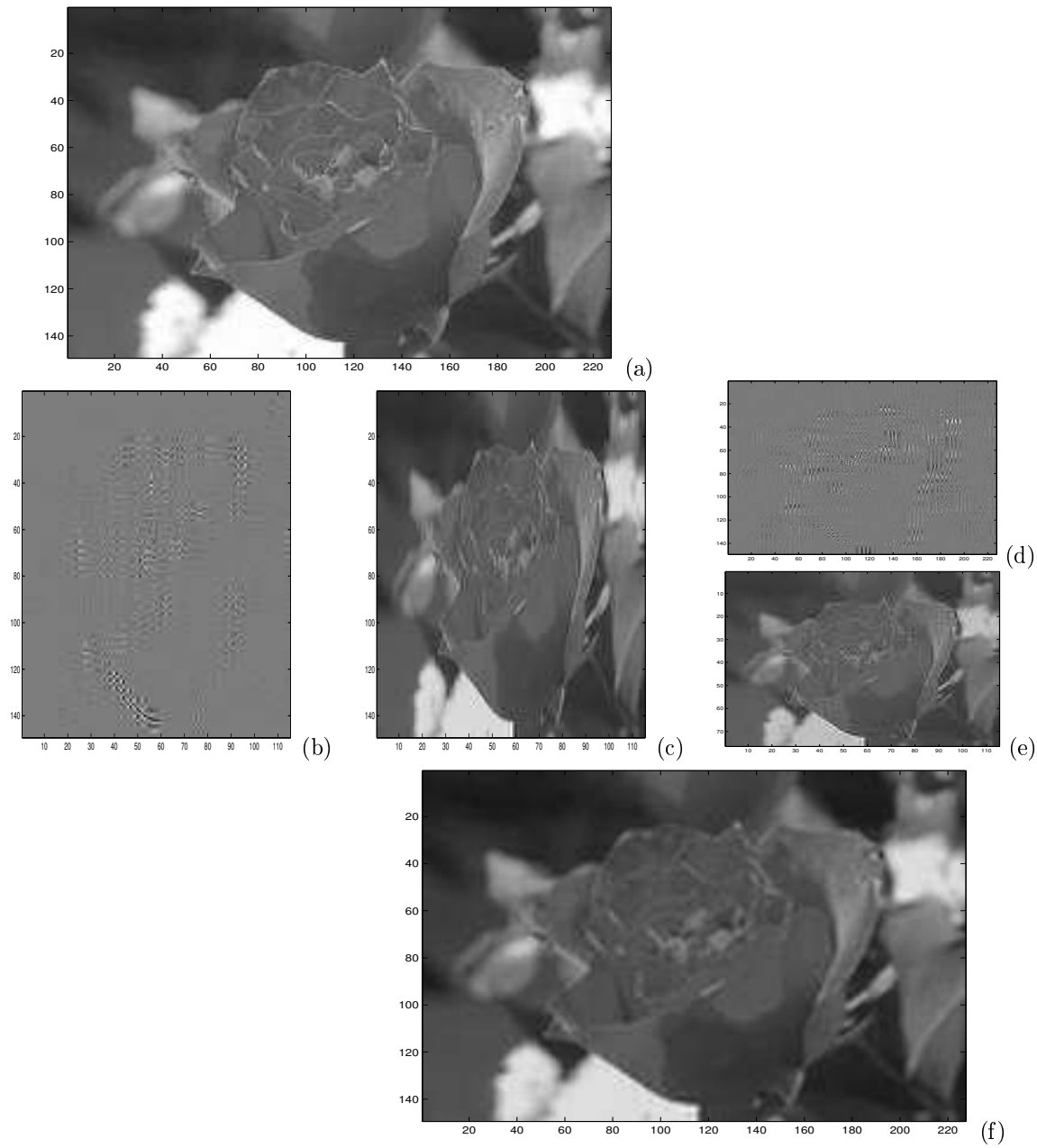
در مبحث فشرده‌سازی تصویر، بعد از تجزیه تصویر به داده‌های تقریب زننده و خطا، داده‌های خطایی را که بر اساس برخی معیارها قابل اغماض و نزدیک به صفر است، کنار می‌گذاریم، و برای

ذخیره‌سازی تصویر تنها داده‌های تقریب زننده و داده‌های خطایی را که حاوی مقادیر قابل توجهی هستند، ذخیره می‌کنیم. معمولاً در تصاویر، لبهٔ ناحیه‌هایی که دارای رنگهای کاملاً متفاوت هستند، به عنوان داده‌های خطا ذخیره می‌شوند، و بدین ترتیب حجم بزرگی از داده‌های خطا مربوط به داخل نواحی پیوستهٔ تصویر قابل اغماض است.

به همین ترتیب، می‌توان با اعمال تغییرات مطلوب بر روی این دو بخش از داده‌های تجزیه شده و ترکیب مجدد آنها، برخی مفاهیم پردازش تصاویر را در عمل پیاده‌سازی کرد. مثلاً با صرفنظر کردن از داده‌های خطا و ترکیب داده‌های تقریب‌زننده با خطای صفر، می‌توانیم وضوح تصویر در لبه‌ها را محو کنیم. و یا مثلاً با ترکیب داده‌های خطا با داده‌های یک تصویر کاملاً سیاه، تصویری از مرزهای تصویر اولیه را بدست آوریم.

به عنوان مثال، در شکل ۵—۵، تصویر اولیهٔ یک گل با ابعاد ۱۵۰ × ۲۳۰ پیکسل (a) به یک تصویر با ابعاد ۷۵ × ۱۱۵ پیکسل (e) فشرده شده است. برای این کار، بعد از مدلسازی اولیهٔ تصویر به وسیلهٔ یک رویهٔ Bـاسپلاین مرتبهٔ $k = 4$ (در هر دو جهت) روی دنبالهٔ گره‌های یکنواخت (در هر دو جهت)، ابتدا نیمی از گره‌ها به طوریکی در میان از بین گره‌های منحنی عمودی رویهٔ تصویر حذف شده‌اند، و بدین ترتیب تصویر به دو بخش تقریب‌زننده (c) و خطا (b) (مرز نواحی غیر همرنگ در جهت عمودی) تجزیه می‌شود. سپس با تکرار همین فرایند در جهت افقی روی تصویر (c)، دو تصویر (d) و (e) به ترتیب به عنوان داده‌های تقریب و خطا بدست می‌آیند. در این وضعیت حجم اطلاعات تصویر (e) یک چهارم اطلاعات تصویر اولیه است. تنها با ذخیره‌سازی اطلاعات تصویر (e) و اطلاعات معنی‌دار تصاویر (d) و (b) می‌توان تصویر اولیه را به طور کامل بازسازی نمود. با این‌حال در صورت صرفنظر کردن از کلیهٔ دادهای خطا، تصویر (f) بازسازی می‌شود که با دقت قابل قبولی با تصویر (a) یکسان است.

در آخر یادآور می‌شویم که از مزایای روشهای ترکیب و تجزیهٔ مبتنی بر حذف گره، عدم وابستگی آنها به نوع خاصی از دنبالهٔ گره‌هاست، در صورتی که روشهای ترکیب و تجزیه موجک عموماً روی دنبالهٔ گره‌های یکنواخت در نظر گرفته می‌شوند. این ویژگی در برخی کاربردها مفید واقع می‌شود. در برخی کاربردها، گره‌ها خود جزیی از داده‌های خام هستند، یا در برخی از کاربردهای دیگر وضعیت گره‌ها در ابتدای امر نشان دهندهٔ برخی ویژگیهای هندسی مدل خواهد بود و بهرحال، مطلوبست که روش تجزیه بر روی



شکل ۵—۵: فشرده‌سازی تصویر: (a) تصویر اصلی، (b) داده‌های خطا و (c) داده‌های تقریب‌زننده بعد از حذف نیمی از گره‌ها در جهت عمودی، (d) داده‌های خطا و (e) داده‌های تقریب‌زننده بعد از حذف نیمی از گره‌ها در جهت افقی از تصویر (c)، (f) تصویر بدست آمده از (e) با درج مجدد گره‌های حذف شده، بدون ترکیب با هیچیک از داده‌های خطا.

دنباله گره‌های معینی عمل کند.

۵–۳ تقریب توابع و مسألهٔ بهترین گره‌ها

اگر در حذف گره‌ها از یک منحنی Bـاسپلاین زیاده‌روی کنیم، نتیجه بدست آمده دیگر ضرورتاً یک تقریب نزدیک به منحنی اولیه نخواهد بود، امّا هنوز از برخی دیدگاه‌ها قابل توجه به نظر می‌رسد. در بخش پیش، هدف، استفاده از روشهای تجزیه برای بدست آوردن مدلی برای داده‌های اولیه با حداقل تعداد گره‌های ممکن بود. به عبارت دیگر، در آنجا به محض گذر از وضعیتی که خطای بین منحنی اولیه و نهایی از میزان قابل چشم‌پوشی فراتر می‌رود، روند تکرار الگوریتم را متوقف می‌کنیم. امّا در اینجا برعکس، اینقدر به عمل حذف گره ادامه می‌دهیم تا تعداد گره‌های باقی مانده تنها برای نمایش برخی توابع خیلی ساده، مثل چندجمله‌ایها و یا برخی اسپلاینها، مناسب باشد.

این یک رهیافت از دیدگاه مباحث طراحی هندسی برای مسألهٔ بهترین تقریب توابع است. فرض کنید که می‌خواهیم بهترین تقریب یک تابع $f \in C^۲[a\ b]$ را بر روی فضای چندجمله‌ایهای مرتبه k پیدا کنیم. با انتخاب n بزرگ ($n \sim ۱۰۰۰$) و دنبالهٔ گره‌های یکنواخت با گره‌های با تکرار k در ابتدا و انتها بطوریکه $a = u_۱ = \cdots = u_k \leq u_{k+۱} \leq \cdots \leq u_n \leq u_{n+۱} = \cdots = u_{n+k} = b$ ، تابع Bـاسپلاین $V_f(t)$ را به صورت

$$V_f(t) = \sum_{i=۱}^{n} f(\xi_i) B_i^k(t)$$

بر روی دنبالهٔ مذکور تعریف می‌کنیم. تابع $V_f(t)$ اصطلاحاً به عنوان تقریب شوئنبرگ تابع f شناخته می‌شود. برای n به قدر کافی بزرگ و $\|f''\|_{L^\infty} < \infty$، تابع $V_f(t)$ یک تقریب بسیار نزدیک به f است.

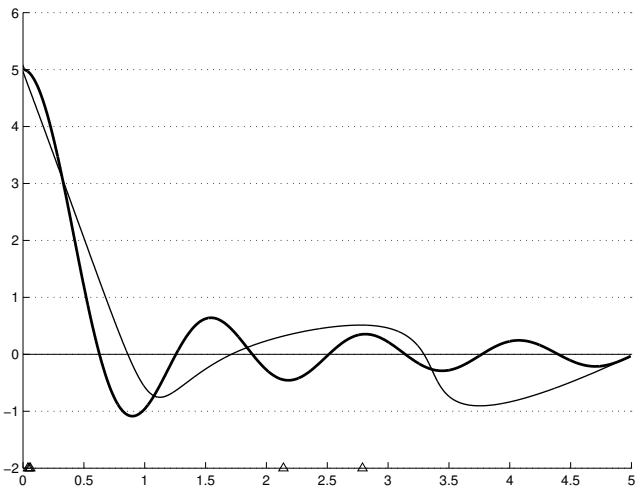
با حذف تمام گره‌های درونی برای $V_f(t)$، به یک تابع Bـاسپلاین بر روی دنباله گره‌های $a = u_۱ = \cdots = u_k, u_{n+۱} = \cdots = u_{n+k} = b$ می‌رسیم. این تابع دقیقاً یک چندجمله‌ای از مرتبهٔ k است و هرچند بهترین تقریب تابع f نیست، ولی به بهترین تقریب آن نزدیک است. این موضوع از آن جهت قابل اعتناست که بیاد آوریم محاسبهٔ بهترین تقریب تابع دلخواه f در نرم L^∞ مستلزم روشهای پیچیده و

محاسبات زیاد است.

همین عملیات وقتی می‌خواهیم بهترین تقریب f را در فضای اسپلاینهای مرتبه k بدست آوریم، قابل تکرار است. منتها مسأله‌ای که عمدتاً در همین رابطهٔ مطرح می‌شود اینست که در بین تمام دنباله گره‌های تعریف شده بر بازهٔ $[a\,b]$ با تعداد ثابتی از گره‌ها l، بهترین تقریب تابع f در فضای اسپلاینهای مرتبهٔ k بر روی کدامیک از دنباله گره‌ها دارای کمترین خطاست؟ این مسأله به عنوان مسألهٔ بهترین گره‌ها[11] و نیز تقریب اسپلاین با گره‌های آزاد[12] شناخته می‌شود.

حل دقیق و تحلیلی این مسأله به برخی مسائل پیچیده در نظریه تقریب منجر می‌شود که به طور عملی قابل پیاده‌سازی نیستند[۳۰]. درعوض برخی روشهای تقریبی که عمدتاً با دیدگاههای طراحی هندسی بیان می‌شوند، برای حل عملی این مسأله وجود دارند[۳۸]، [۴۲] و [۱۳].

در اینجا نیز می‌توان با شروع از دنبالهٔ بزرگی از گره‌ها و حذف آنها تا زمانیکه فقط l گره باقی مانده‌است، بر روی یک مدل اولیه از تابع (مثل تقریب شوئنبرگ) به یک جواب نزدیک به جواب دقیق مسألهٔ بهترین گره‌ها رسید.

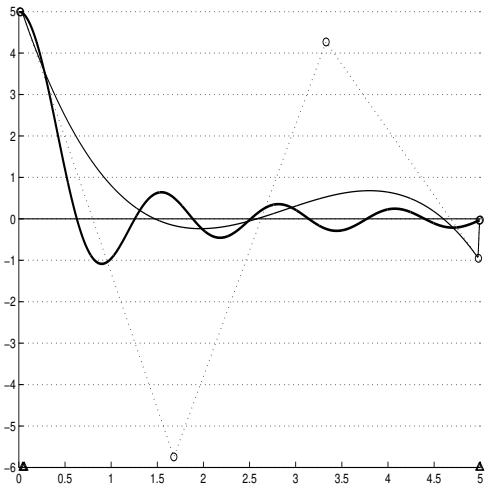


شکل ۵—٦: تقریب تابع بوسیلهٔ Bـاسپلاین مکعبی: تابع $\frac{1}{x}\sin(5x)$ (منحنی کلفت)، تابع Bـاسپلاین (منحنی نازک). توجه کنید که دو گره داخل دامنه باقی‌مانده است.

در شکل ۵—٦، با انتخاب ۴۸۸ گره بروش ارزیابی مجدد از بین ۵۰۰ گره اولیه یکنواخت و حذف

[11] Good knots

[12] Free knots



شکل ۵—۷: تقریب تابع بوسیلهٔ چندجمله‌ای درجه سه: چندضلعی کنترلی تابع Bـاسپلاین با نقطه‌چین نشان داده شده‌است. در اینجا تمام گره‌ها بجز چند گره چسبیده به لبه‌های دامنه، حذف شده‌اند.

آنها با بهترین تقریب L_∞ پیوسته، بر روی تقریب شوئنبرگ تابع $\frac{1}{x}\sin(5x)$ (بعنوان تقریب اولیه)، به یک تابع Bـاسپلاین مرتبه ۴ می‌رسیم که به عنوان یک تقریب اسپلاین برای این تابع مناسب به نظر می‌رسد. اختلاف L_∞ این دو تابع برابر ۱٫۳۱۰۴ است که همانطور که در شکل دیده می‌شود تقریباً مشابه خطای بهترین تقریب (کلی) یک تابع، متناوباً بین تابع اصلی و تقریب آن روی می‌دهد.

به همین ترتیب در شکل ۵—۷، با انتخاب همین تعداد گره ولی بروش یک بار مرتب‌سازی و حذف آنها با تقریب L_∞ گسسته، به یک تابع Bـاسپلاین مرتبه ۴ می‌رسیم که چون تمام گره‌های درونی آن بجز چند گره بسیار نزدیک به لبه‌های دامنه حذف شده‌اند، به عنوان یک تقریب چندجمله‌ای (درجه سه) برای تابع مذکور مناسب به نظر می‌رسد. خطای این شبه تقریب برابر ۲٫۳۱۰۰ است.

پیوست A

برنامه‌هایی به زبان MATLAB

برنامه‌هایی که در این فصل ارائه می‌شوند، در اصل مجموعه‌ای از توابع مستقل بر اساس زبان برنامه‌نویسی MATLAB است. نرم‌افزار ریاضی MATLAB یک برنامهٔ قوی برای حل مسائل ریاضی به خصوص عددی است و استفاده از آن امروزه بسیار متداول است. این نرم‌افزار علاوه بر محیط خط فرمانی[1] دارای یک محیط برنامه‌نویسی قوی نیز است. علت انتخاب این زبان، علاوه بر سادگی و ایجاز آن، در اینست که دستورالعملهای آن حتیٰ برای اشخاص ناآشنا به MATLAB نیز قابل فهم و قابل پیاده‌سازی به دیگر زبانهای برنامه‌نویسی است.

برنامه‌هایی که در ادامه می‌آیند، به طور کامل بارها اجرا و خطایابی شده‌اند و تمام جداول و اشکال این رساله به وسیلهٔ آنها بدست آمده‌اند. نسخهٔ نرم‌افزار MATLAB بکار رفته 5.3 است که تحت سیستم عامل ویندوز اجرا می‌شود.

نام اکثر متغیرهای بکار رفته در برنامه‌های ذیل با کمی ساده‌سازی مطابق با نمادگذاریهای بکار رفته در متن است و در اینجا بدون ذکر جزئیات برنامه‌نویسی، تنها به نوع و مفهوم هر یک از پارامترهای ورودی و خروجی توابع پرداخته می‌شود. به طور کلی در برنامه‌های زیر، n نشاندهندهٔ تعداد رئوس کنترلی، c یک بردار سطری شامل رئوس کنترلی و به طور دقیقتر برای یک منحنی Bـاسپلاین در $\mathbb{R}^S$، بردار c به صورت یک ماتریس $S\times$n است. k مثل همیشه مرتبهٔ Bـاسپلاین، u یک بردار سطری حاوی دنبالهٔ گره‌هاست و delta و mu نشاندهندهٔ مکان درج (اندیس) و مرتبهٔ تکرار گره جدیدی (موجودی) است که می‌خواهیم درج

[1] Command line

(حذف) کنیم.

توجه. برای هر یک از توابع زیر یک فایل جداگانه با همان نام و با پسوند m. بسازید و در یک شاخهٔ مستقل در مجموعهٔ کتابخانهٔ MATLAB قرار دهید و مسیر این شاخه را به مسیرهای جستجوی MATLAB اضافه کنید. در صورت نیاز به جزئیات بیشتر به راهنمای MATLAB رجوع کنید.

جستجوی سریع مکان و مرتبهٔ تکرار گره

ورودی. n, k, u, t, a, b .

t یک بردار سطری حاوی مقادیر برای پارامتر یا گره‌هایی است که می‌خواهیم حذف یا درج کنیم.

a و b هر کدام بردارهای سطری متناظر با t و حاوی اعداد صحیحی هستند به طوری که تابع ffknot مکان t را در بازهٔ [u(a) u(b)] مورد جستجو قرار می‌دهد.

خروجی. delta, mu .

delta بردار متناظر با اندیس درج هر یک از عناصر t است.

mu بردار متناظر با مرتبهٔ تکرار هر یک از عناصر t است.

```
function [delta, mu]=ffknot(n, k, u, t, a, b)
% Fast Find Knot: [delta, mu]=ffknot(n, k, u, t, a, b)
j=length(t);
while j>0
   [delta(j), mu(j)]=affknot(n, k, u, t(j), a(j), b(j));
   j=j-1;
end

function [delta, mu]=affknot(n, k, u, t, a, b)
% Atomic Fast Find Knot: [delta, mu]=affknot(n, k, u, t, a, b)
if t<u(a)
   delta=a-1;
   mu=0;
elseif u(b)<=t
   delta=b;
   mu=0;
   i=b;
   while t==u(i)
      mu=mu+1;
      i=i-1;
      if i<1
```

```
         break;
      end
   end
else
   while b-a>1
      c=floor((a+b)/2);
      if u(c)<=t
         a=c;
      else
         b=c;
      end
   end
   delta=a;
   mu=0;
   i=a;
   while t==u(i)
      mu=mu+1;
      i=i-1;
      if i<1
         break;
      end
   end
end
```

ارزیابی جدول توابع پایهٔ Bـاسپلاین

ورودی. n,k,u,t,delta,mu .

t, delta, mu همانند پارامترهای ffknot تعریف می‌شوند.

خروجی. table یک آرایهٔ سه بعدی با ابعاد k×k×size(t) است که هر ماتریس پائین مثلثی tabel(:,:,j) حاوی مقادیر توابع پایه بی‌اسپلاین مرتبهٔ $1,\ldots,k$ غیر صفر به ازای t(j) است.

```
function table=bstable(n,k,u,t,delta,mu)
% Evaluate BSpline Table at t:  table=bstable(n,k,u,t,delta,mu)
j=length(t);
while j>0
   table(:,:,j)=abstable(n,k,u,t(j),delta(j),mu(j));
   j=j-1;
end

function table=abstable(n,k,u,t,delta,mu)
% Atomic Evaluate BSpline Table at t:  table=abstable(n,k,u,t,delta,mu)
table=zeros(k,k);
table(k,1)=1;
for r=1:k-1
```

```
   i=delta-r+1;
   for j=k-r+1:k
      w=(t-u(i))/(u(i+r)-u(i));
      table(j-1,r+1)=table(j-1,r+1)+(1-w)*table(j,r);
      table(j,r+1)=w*table(j,r);
      i=i+1;
   end
end
```

ارزیابی مقادیر توابع پایهٔ Bـاسپلاین

ورودی. n,k,u,t,delta,mu همانند پارامترهای bstable تعریف می‌شوند.

خروجی. bsval یک ماتریس با ابعاد k×size(t) است که هر بردار ستونی آن حاوی مقادیر توابع پایهٔ Bـاسپلاین مرتبهٔ k غیر صفر متناظر با مقادیر پارامتر t است.

```
function bv=bsval(n,k,u,t,delta,mu)
% Evaluate BSpline bases values bv at t:  bv=bsval(n,k,u,t,delta,mu)
j=length(t);
while j>0
   bv(:,j)=absval(n,k,u,t(j),delta(j),mu(j));
   j=j-1;
end

function bv=absval(n,k,u,t,delta,mu)
% Atomic Evaluate BSpline bases values bv at t:  bv=absval(n,k,u,t,delta,mu)
bv=zeros(k,1);
bv(k)=1;
for r=1:k-1
   i=delta-r+1;
   for j=k-r+1:k
      w=(t-u(i))/(u(i+r)-u(i));
      bv(j-1)=bv(j-1)+(1-w)*bv(j);
      bv(j)=w*bv(j);
      i=i+1;
   end
end
```

ترکیب توابع پایهٔ Bـاسپلاین

ورودی. c,n,k,u,t,delta,mu همانطور که در بالا ذکر شد، تعریف می‌شوند.

خروجی. bc یک بردار سطری حاوی مقادیر تابع یا منحنی Bـاسپلاین تعریف شده با ضرایب کنترلی c، به ازای هر یک از مقادیر پارامتر در t است. تعداد ستونهای آن متناظر یا بعد فضایی است که عناصر c

در آن قرار دارند.

```
function bc=bscomb(c,n,k,u,t,delta,mu)
% Combination of BSpline bases:  bc=bscomb(c,n,k,u,t,delta,mu)
mu=min(mu, k-1);
j=length(t);
while j>0
   bc(:,j)=abscomb(c(:,delta(j)-k+1:delta(j)-mu(j)),n,k,u,t(j),delta(j),mu(j));
   j=j-1;
end

function bc=abscomb(c,n,k,u,t,delta,mu)
% Atomic Combination of BSpline bases:  bc=abscomb(c,n,k,u,t,delta,mu)
r=k-mu;
for j=2:r
   i=delta-mu;
   k=k-1;
   for l=r:-1:j
      w=(t-u(i))/(u(i+k)-u(i));
      c(:,l)=w*c(:,l)+(1-w)*c(:,l-1);
      i=i-1;
   end
end
bc=c(:,r);
```

محاسبهٔ ضرایب Bـاسپلاین مشتق

ورودی. `c, n, k, u` .

خروجی. `dc` بردار سطری حاوی ضرایب تابع Bـاسپلاین مشتق متناظر با `c` است.

```
function dc=dbscoef(c, n, k, u)
% Calculate Coeficient of Derivative BSpline curve:  dc=dbscoef(c, n, k, u)
[m n]=size(c);
o=zeros(m,1);
dc=[c o]-[o c];
n=n+1;
k=k-1;
for i=1:n
   if u(i+k)>u(i)
      dc(:,i)=k*dc(:,i)/(u(i+k)-u(i));
   else
      dc(:,i)=o;
   end
end
```

محاسبهٔ نقاط گرویل

ورودی. n,k,u .

خروجی. xi بردار سطری حاوی نقاط گرویل متناظر با دنباله گره‌های u.

```
function xi=grev(n,k,u)
% Calculate Greville abscissae:  xi=grev(n,k,u)
xi=u(2:n+1);
for i=2:k-1
   xi=xi+u(i+1:i+n);
end
xi=xi/(k-1);
```

درج تک گره

ورودی. c,n,k,u,t,delta,mu .

خروجی. c, n, u مقادیر جدید پارامترها بعد از درج گره t.

```
function [c, n, u]=insknot(c,n,k,u,t,delta,mu)
% Insert knot t in u and refine control polygon c:
%   [c n u]=insknot(c,n,k,u,t,delta,mu)
if mu<k
   c=[c(:,1:delta-mu) c(:,delta-mu:n)];
   k=k-1;
   for i=delta-mu:-1:delta-k+1
      w=(t-u(i))/(u(i+k)-u(i));
      c(:,i)=w*c(:,i)+(1-w)*c(:,i-1);
   end
   n=n+1;
   u=[u(1:delta) t u(delta+1:n+k)];
end
```

درج مکرر یک گره

ورودی. c,n,k,u,t,delta,mu,r .

r یک عدد درست که تعداد مرتبهٔ تکرار درج گره t را نشان می‌دهد.

خروجی. c, n, u مقادیر پارامتر بعد از r بار درج گره t در u.

```
function [c, n, u]=rinsknot(c,n,k,u,t,delta,mu,r)
% Repeat Inserting knot t in u, r times and refine control polygon c:
```

```
%   [c n u]=rinsknot(c,n,k,u,t,delta,mu,r)
r=min(r,k-mu);
[l n]=size(c);
c=[c(:,1:delta-mu) zeros(l,r) c(:,delta-mu+1:n)];
for j=r:-1:1
   c(:,delta-mu+j)=c(:,delta-mu);
   k=k-1;
   for i=delta-mu:-1:delta-k+1
      w=(t-u(i))/(u(i+k)-u(i));
      c(:,i)=w*c(:,i)+(1-w)*c(:,i-1);
   end
end
n=n+r;
u=[u(1:delta) t*ones(1,r) u(delta+1:n+k)];
```

حذف گره مجازی

ورودی. c, n, k, u, delta, mu .

خروجی. c1, c2 دو بردار سطری بطول n-1، حاوی چند ضلعیهای کنترلی حاصل از حذف مجازی گره u(delta) با مؤلفه‌های همنوع با مؤلفه‌های c.

```
function [c1, c2]=pseudorem(c, n, k, u, delta, mu)
% Pseudo Knot Removal:
%   [c1, c2]=pseudorem(c, n, k, u, delta, mu)
t=u(delta);
c1=c(:,delta-k:delta-mu);
c2=c(:,delta-k+1:delta-mu+1);

j=delta-k+1;
for i=2:k-mu+1
   v=(u(j+k)-u(j))/(t-u(j));
   c1(:,i)=v*c1(:,i)+(1-v)*c1(:,i-1);
   j=j+1;
end

j=delta-mu;
for i=k-mu:-1:1
   v=(u(j+k)-u(j))/(u(j+k)-t);
   c2(:,i)=v*c2(:,i)+(1-v)*c2(:,i+1);
   j=j-1;
end
```

ارزیابی ضرایب براکتی

ورودی. n,k,u,delta,mu .

mu و delta همانند پارامترهای تابع bstable تعریف می‌شوند.

خروجی. ba یک ماتریس با ابعاد (k+1)×size(delta) که هر ستون آن حاوی ضرایبی براکتی متناظر با حذف هر گره u(delta) است. در این تابع تنها ضرایب براکتی $\begin{bmatrix} k-1 \\ i \end{bmatrix}$ به ازای i=mu,...,k+1 محاسبه می‌شوند و در هر ستون ba ضرایب متناظر با i=1,...,mu-1 صفر قرار داده می‌شوند.

```
function ba=bracket(n,k,u,delta,mu)
% Evaluate BRACKET coeficient [k-1, i] ,for i= mu .. k+1:
%   ba=bracket(n,k,u,delta,mu)
j=length(delta);
while j>0
   ba(:,j)=abracket(n,k,u,delta(j),mu(j));
   j=j-1;
end

function ba=abracket(n,k,u,delta,mu)
ba=zeros(k+1,1);
t=u(delta);
ba(k+1)=1; v=1;
j=delta-mu;
for i=k:-1:mu
   w=(t-u(j))/(u(j+k)-u(j));
   ba(i)=v/w*ba(i+1);
   v=1-w;
   j=j-1;
end
```

ارزیابی ضریب خطای حذف گره

ورودی. c,n,k,u,delta,mu .

mu و delta همانند پارامترهای تابع bstable تعریف می‌شوند.

خروجی. d یک بردار سطری حاوی ضرایب خطای حذف هر یک از گره‌های u(delta) است. نوع مؤلفه‌های آن همنوع مؤلفه‌های c است.

```
function d=difrem(c,n,k,u,delta,mu)
% Evaluate Difference between knot removed BSpline curve and original curve :
%   d=difrem(c,n,k,u,delta,mu)
```

```
alt=(-1).^[-k:0]';
ba=bracket(n,k,u,delta,mu);
j=length(delta);
while j>0
   dd=delta(j); mm=mu(j);
   d(:,j)=c(:,dd-k:dd-mm+1)*(alt(mm:k+1).*ba(mm:k+1,j));
   j=j-1;
end
```

محاسبهٔ مقادیر مناسب برای m_1, m_2

ورودی. n,k,delta,mu .

خروجی. m1, m2 مقادیر مناسب برای m_1, m_2 مورد نیاز در الگوریتمهای حذف گره و بازسازی، متناظر با حذف گره u(delta).

```
function [m1, m2]=suitm(n,k,delta,mu)
% Calculate suitable m1 and m2 for RemKnots and Reconsts ,too.
%   [m1, m2]=suitm(n,k,delta,mu)
z=ceil(k/2)+1;
m1=max(0, k-delta+z);
m2=max(0, delta-mu-n+z);
```

حذف گره ساده

ورودی. c,n,k,u,delta,mu,m1,m2 .

m1, m2 همانند پارامترهای خروجی تابع suitm تعریف می‌شوند.

خروجی. c, n, u, a مقادیر پارامتر بعد از حذف تک گره u(delta) بدون جستجوی هیچ تقریبی.

در اینجا مقدار بازگشتی a همیشه برابر 1 است.

```
function [c, n, u, a]=remknot(c,n,k,u,delta,mu,m1,m2)
% Knot Removal in a Simple method:
%  remknot(c,n,k,u,delta,mu,m1,m2)
[c1, c2]=pseudorem(c,n,k,u,delta,mu);
m=k-mu+1-m1-m2;
l=[1:m]/(m+1);
cc=[ ];
for i=1:m
   cc(:,i)=(1-l(i))*c1(:,i+m1)+l(i)*c2(:,i+m1);
end
c=[c(:,1:delta-k-1) c1(:,1:m1) cc c2(:,m+m1+1:k-mu+1) c(:,delta-mu+2:n)];
```

```
u=[u(1:delta-1) u(delta+1:n+k)];
n=n-1;
a=1;
```

حذف گره با تقریب L_∞ گسسته

ورودی. c,n,k,u,delta,mu,m1,m2 همانند پارامترهای تابع remknot تعریف می‌شوند.

خروجی. c, n, u, a مقادیر پارامتر بعد از حذف تک گره u(delta) با تقریب L_∞ گسسته.

a$\|D\|_2$ کران بالای خطای حذف گره، بدست آمده از این روش تقریب است.

```
function [c, n, u, a]=remknotmax(c,n,k,u,delta,mu,m1,m2)
% Knot Removal in Discrete L_infty method:
%  remknotmax(c,n,k,u,delta,mu,m1,m2)
[c1, c2]=pseudorem(c,n,k,u,delta,mu);
g=bracket(n,k,u,delta,mu);
g=cumsum(g(mu+m1:k+1-m2));
m=k-mu+1-m1-m2;
l=g(1:m)/g(m+1);
cc=[ ];
for i=1:m
   cc(:,i)=(1-l(i))*c1(:,i+m1)+l(i)*c2(:,i+m1);
end
c=[c(:,1:delta-k-1) c1(:,1:m1) cc c2(:,m+m1+1:k-mu+1) c(:,delta-mu+2:n)];
u=[u(1:delta-1) u(delta+1:n+k)];
n=n-1;
a=1/g(m+1);
```

حذف گره با تقریب L_2 گسسته

ورودی. c,n,k,u,delta,mu,m1,m2 همانند پارامترهای تابع remknot تعریف می‌شوند.

خروجی. c, n, u, a مقادیر پارامتر بعد از حذف تک گره u(delta) با تقریب L_2 گسسته.

a$\|D\|_2$ کران بالای خطای حذف گره، بدست آمده از این روش تقریب است.

```
function [c, n, u, a]=remknotl2(c,n,k,u,delta,mu,m1,m2)
% Knot Removal in Discrete L_2 method:
%   remknotl2(c,n,k,u,delta,mu,m1,m2)
[c1, c2]=pseudorem(c,n,k,u,delta,mu);
g=bracket(n,k,u,delta,mu);
ag=g(mu+m1:k+1-m2);
g=cumsum(ag.^2);
m=k-mu+1-m1-m2;
```

```
l=g(1:m)/g(m+1);
cc=[ ];
for i=1:m
   cc(:,i)=(1-l(i))*c1(:,i+m1)+l(i)*c2(:,i+m1);
end
c=[c(:,1:delta-k-1) c1(:,1:m1) cc c2(:,m+m1+1:k-mu+1) c(:,delta-mu+2:n)];
u=[u(1:delta-1) u(delta+1:n+k)];
n=n-1;
a=max(ag)/g(m+1);
```

محاسبهٔ ضرایب اسپلاین چبیشفی

ورودی. n,k,u,delta,mu,m1,m2,it همانند پارامترهای تابع remknot تعریف می‌شوند.

it تعداد تکرارهای ثابت الگوریتم رمز (معمولاً با $k-۱ \leq$it$\leq k+۱$ فراخوانی می‌شود).

خروجی. h بردار سطری حاوی ضرایب اسپلاین چبیشفی وابسته به حذف گره u(delta).

```
function h=chebyspline(n,k,u,delta,mu,m1,m2,it)
% Compute Chebyshevian Spline coeficients. it is difference error
% between the knot removed BSpline and original BSpline:
%   h=chebyspline(n,k,u,delta,mu,m1,m2,it)
o=zeros(1,k);
m=k-mu+1-m1-m2;
alt=(-1).^[m2-m:m2];
if k<3
   h=alt;
   return;
end
n1=delta-k+m1;  n2=delta-mu+1-m2;
xi=grev(m+1,k,u(n1:n2+k));
e=xi;

while it>0
   % 1,-1 interpolation
   [de me]=ffknot(n,k,u,e,[n1:n2],[n1+k:n2+k]);
   b=bstable(n,k,u,e,de,me);
   a=zeros(m+1);
   for i=1:m+1
      di=de(i)-n1-k+1;
      if di>=0
         ab=1;
         aa=di+1;
      else
         ab=1-di;
         aa=1;
      end
```

```
      di=n2-de(i);
      if di>=0
         bb=k;
         ba=m+1-di;
      else
         bb=k+di;
         ba=m+1;
      end
      a(aa:ba,i)=b(ab:bb,k,i);
   end
   % Solve linear system hA=alt
   h=alt/a;

   % One Step One Itration Newton method for approach to Extermal points
   gc=dbscoef(h,m+1,k,u(n1:n2+k));
   hc=dbscoef(gc,m+2,k-1,u(n1:n2+k));
   gc=[o gc o];
   hc=[o hc o];
   j=k-delta-m1+1;
   for i=1:m+1
      if me(i)<k-1
         ge(i)=gc(de(i)+2+j:de(i)+k+j)*squeeze(b(2:k,k-1,i));
         he(i)=hc(de(i)+3+j:de(i)+k+j)*squeeze(b(3:k,k-2,i));
         ee=e(i)-ge(i)/he(i);
         if i>1
            ea=e(i-1);
         else
            ea=u(n1);
         end
         if i<=m
            eb=e(i+1);
         else
            eb=u(n2+k);
         end

         if (alt(i)*he(i)>0) | (ee>eb) | (ee<ea)
            e(i)=xi(i)+(alt(i)-h(i))*(1/gc(i+k)+1/gc(i+k+1))/2;
         else
            e(i)=ee;
         end
      end
   end
   it=it-1;
end
```

حذف گره با بهترین تقریب L_∞ پیوسته

ورودی. c,n,k,u,delta,mu,m1,m2 همانند پارامترهای تابع remknot تعریف می‌شوند.

خروجی. c, n, u, a مقادیر پارامتر بعد از حذف تک گره u(delta) با بهترین تقریب L_∞ پیوسته.

a$\|D\|_\infty$ خطای حذف گره، بدست آمده از این روش تقریب است.

```
function [c, n, u, a]=remknotsup(c,n,k,u,delta,mu,m1,m2)
% Knot Removal in Continuous L_infty method:
%   remknotsup(c,n,k,u,delta,mu,m1,m2)
[c1, c2]=pseudorem(c,n,k,u,delta,mu);
g=bracket(n,k,u,delta,mu);
h=chebyspline(n,k,u,delta,mu,m1,m2,k);
g=cumsum(g(mu+m1:k+1-m2).*abs(h'));
m=k-mu+1-m1-m2;
l=g(1:m)/g(m+1);
cc=[ ];
for i=1:m
   cc(:,i)=(1-l(i))*c1(:,i+m1)+l(i)*c2(:,i+m1);
end
c=[c(:,1:delta-k-1) c1(:,1:m1) cc c2(:,m+m1+1:k-mu+1) c(:,delta-mu+2:n)];
u=[u(1:delta-1) u(delta+1:n+k)];
n=n-1;
a=1/g(m+1);
```

بازسازی منحنی اولیه بعد از حذف گره L_∞ گسسته

ورودی. c,n,k,u,t,delta,mu,d .

d ضریب خطای گره حذف شدهٔ t.

خروجی. c, n, u مقادیر پارامتر بعد از درج گره t و ترکیب با d.

```
function [c, n, u]=reconstmax(c,n,k,u,t,delta,mu,d)
% Reconstruction Discrete L_infty Knot Removed BSpline by Knot inserting
% then error composition:
%   [c, n, u]=reconstmax(c,n,k,u,t,delta,mu,d)
[c n u]=insknot(c,n,k,u,t,delta,mu);
delta=delta+1;
mu=mu+1;
[m1 m2]=suitm(n,k,delta,mu);
m=k-mu+1;
g=bracket(n,k,u,delta,mu);
ad=d/sum(g(mu+m1:k+1-m2));
h=(-1).^[-m:0];
c(:,delta-k+m1:delta+1-mu-m2)=c(:,delta-k+m1:delta+1-mu-m2)+ad*h(1+m1:m+1-m2);
```

بازسازی منحنی اولیه بعد از حذف گره L_2 گسسته

ورودی. c,n,k,u,t,delta,mu,d همانند پارامترهای تابع reconstmax تعریف می‌شوند.

خروجی. c, n, u همانند پارامترهای تابع reconstmax تعریف می‌شوند.

```
function [c, n, u]=reconstl2(c,n,k,u,t,delta,mu,d)
% Reconstruction Discrete L_2 Knot Removed BSpline by Knot inserting
% then error composition:
%   [c, n, u]=reconstl2(c,n,k,u,t,delta,mu,d)
[c n u]=insknot(c,n,k,u,t,delta,mu);
delta=delta+1;
mu=mu+1;
[m1 m2]=suitm(n,k,delta,mu);
m=k-mu+1;
g=bracket(n,k,u,delta,mu);
h=(-1).^[-m:0];
h=h(1+m1:m+1-m2).*g(mu+m1:k+1-m2)';
ad=d/(h*h');
c(:,delta-k+m1:delta+1-mu-m2)=c(:,delta-k+m1:delta+1-mu-m2)+ad*h;
```

بازسازی منحنی اولیه بعد از حذف گره L_∞ پیوسته

ورودی. c,n,k,u,t,delta,mu,d همانند پارامترهای تابع reconstmax تعریف می‌شوند.

خروجی. c, n, u همانند پارامترهای تابع reconstmax تعریف می‌شوند.

```
function [c, n, u]=reconstsup(c,n,k,u,t,delta,mu,d)
% Reconstruction Continuous L_infty Knot Removed BSpline by Knot inserting
% then error composition:
%   [c, n, u]=reconstsup(c,n,k,u,t,delta,mu,d)
[c n u]=insknot(c,n,k,u,t,delta,mu);
delta=delta+1;
mu=mu+1;
[m1 m2]=suitm(n,k,delta,mu);
h=chebyspline(n,k,u,delta,mu,m1,m2,k);
g=bracket(n,k,u,delta,mu);
ad=d/(abs(h)*g(mu+m1:k+1-m2));
c(:,delta-k+m1:delta+1-mu-m2)=c(:,delta-k+m1:delta+1-mu-m2)+ad*h;
```

در ادامه چند مورد برنامه که نحوهٔ بکارگیری توابع اساسی فوق را برای نمایش برخی کاربردها نشان می‌دهند، بعنوان نمونه ارائه شده‌اند.

رسم منحنی Bـاسپلاین بهمراه چندضلعی کنترلی آن

ورودی. c,n,k,u .

خروجی. ندارد.

```
function bsdemo(c,n,k,u)
% Demonstration of BSpline curve (function) and its Control Polygon
% (deBoor Polygon):
%   bsdemo(c,n,k,u)
[l n]=size(c);
t=u(k):0.05:u(n+1)-0.001;
m=length(t);
[delta mu]=ffknot(n,k,u,t,ones(1,m),(n+k)*ones(1,m));
bc=bscomb(c,n,k,u,t,delta,mu);
hold on
cla
if l==1
   plot(t,bc);
   xi=grev(n,k,u);
   plot(xi,c,'ro:');
elseif l==2
   plot(bc(1,:),bc(2,:));
   plot(c(1,:),c(2,:),'ro:');
end
hold off
```

تجزیه منحنی Bـاسپلاین با انتخاب گره با ارزیابی مجدد

ورودی. c,n,k,u,met,p .

met یک رشته حرفی شامل دو نویسه است. نویسهٔ اول نوع نرمی را که در فرایند تجزیه برای مسألهٔ تقریب در حذف گره بکار می‌رود، مشخص می‌کند و می‌تواند یکی از نویسه‌های '2' برای نرم L_2 گسسته، 'm' برای نرم L_∞ گسسته و یا 's' برای نرم L_∞ پیوسته باشد. نویسهٔ دوم، مفهوم پارامتر p را برای برنامه مشخص می‌کند.

اگر کاراکتر نویسهٔ برابر 'n' باشد، آنگاه روند حذف گره تا رسیدن تعداد داده‌ها به p ادامه می‌یابد، و اگر برابر 'r' باشد، روند حذف گره دقیقاً p بار تکرار می‌شود،

و اگر برابر 'e' باشد، آنگاه روند حذف گره تا وقتیکه خطای حذف گره بین دو منحنی متوالی در این فرایند از p تجاوز نکرده است، ادامه می‌یابد.

خروجی. c, n, u, d, r, t . سه پارامتر اول متناظر با داده‌های تقریب زننده و سه پارامتر دوم متناظر با داده‌های خطا بعد از پایان فرایند تجزیه هستند.

```
function [c, n, u, d, r, t]=decompdemo1(c,n,k,u,met,p)
% Demonstration of BSpline Decomposition with Update method:
%   [c, n, u, d, r, t]=decompdemo1(c,n,k,u,met,p)
%   met='xy': x=2 -> l2, x=m -> lmax, x=s -> Lsup
%             y=n -> n=p, y=r -> r=p, y=e -> a||D||<p
d=[ ];
r=0;
t=[ ];
while 1
   ddd=difrem(c,n,k,u,k+1:n,ones(1,n-k));
   [ww di]=min(sum(ddd.^2));
   dd=ddd(:,di);
   d=[d dd];
   di=di+k;
   t=[t u(di)];
   r=r+1;
   [m1 m2]=suitm(n,k,di,1);
   if met(1)=='2'
      [c n u a]=remknotl2(c,n,k,u,di,1,m1,m2);
   elseif met(1)=='m'
      [c n u a]=remknotmax(c,n,k,u,di,1,m1,m2);
   elseif met(1)=='s'
      [c n u a]=remknotsup(c,n,k,u,di,1,m1,m2);
   else
      return;
   end

   if met(2)=='n'
      if n<=p
         break;
      end
   elseif met(2)=='r'
      if r>=p
         break;
      end
   elseif met(2)=='e'
      if a*norm(dd)>p
         break;
      end
   else
```

```
      return
   end
end
```

تجزیه منحنی Bـاسپلاین با انتخاب گره با یکبار مرتب‌سازی

ورودی. c,n,k,u,met,p همانند پارامترهای برنامهٔ decompdemo1 تعریف می‌شوند.

خروجی. c, n, u, d, r, t همانند پارامترهای برنامهٔ decompdemo1 تعریف می‌شوند.

```
function [c, n, u, d, r, t]=decompdemo2(c,n,k,u,met,p)
% Demonstration of BSpline Decomposition with Once Sorting method:
%   [c, n, u, d, r, t]=decompdemo2(c,n,k,u,met,p)
%   met='xy': x=2 -> l2, x=m -> lmax, x=s -> Lsup
%             y=n -> n=p, y=r -> r=p, y=e -> a||D||<p
d=[ ];
r=0;
t=[ ];

ddd=difrem(c,n,k,u,k+1:n,ones(1,n-k));
[ww di]=sort(sum(ddd.^2));
uu=u(di+k);

while 1
   r=r+1;
   di=ffknot(n,k,u,uu(r),1,n+k);
   dd=difrem(c,n,k,u,di,1);
   d=[d dd];
   t=[t u(di)];
   [m1 m2]=suitm(n,k,di,1);
   if met(1)=='2'
      [c n u a]=remknotl2(c,n,k,u,di,1,m1,m2);
   elseif met(1)=='m'
      [c n u a]=remknotmax(c,n,k,u,di,1,m1,m2);
   elseif met(1)=='s'
      [c n u a]=remknotsup(c,n,k,u,di,1,m1,m2);
   else
      return;
   end

   if met(2)=='n'
      if n<=p
         break;
      end
   elseif met(2)=='r'
      if r>=p
         break;
```

```
      end
   elseif met(2)=='e'
      if a*norm(dd)>p
         break;
      end
   else
      return
   end
end
```

تجزیهٔ منحنی Bـاسپلاین با حذف گره‌های از پیش انتخاب‌شده

ورودی. c,n,k,u,t,met,p همانند پارامترهای برنامهٔ decompdemo1 تعریف می‌شوند.

t یک بردار سطری حاوی مقادیر گره‌هایی است که به ترتیب در فرایند تجزیه حذف می‌شوند.

خروجی. c, n, u, d, r همانند پارامترهای خروجی برنامهٔ decompdemo1 هستند، به جز این که در اینجا t به عنوان ورودی است و دیگر نیازی به برگرداندن آن نیست.

```
function [c, n, u, d, r]=decompdemo3(c,n,k,u,t,met,p)
% Demonstration of BSpline Decomposition with Before Selected knots:
%   [c, n, u, d, r]=decompdemo3(c,n,k,u,t,met,p)
%   met='xy': x=2 -> l2, x=m -> lmax, x=s -> Lsup
%             y=n -> n=p, y=r -> r=p, y=e -> a||D||<p
d=[ ];
r=0;

while 1
   r=r+1;
   [di ri]=ffknot(n,k,u,t(r),1,n+k);
   dd=difrem(c,n,k,u,di,ri);
   d=[d dd];
   [m1 m2]=suitm(n,k,di,1);
   if met(1)=='2'
      [c n u a]=remknotl2(c,n,k,u,di,1,m1,m2);
   elseif met(1)=='m'
      [c n u a]=remknotmax(c,n,k,u,di,1,m1,m2);
   elseif met(1)=='s'
      [c n u a]=remknotsup(c,n,k,u,di,1,m1,m2);
   else
      return;
   end

   if met(2)=='n'
      if n<=p
         break;
```

```
      end
   elseif met(2)=='r'
      if r>=p
         break;
      end
   elseif met(2)=='e'
      if a*norm(dd)>p
         break;
      end
   else
      return
   end
end
```

ترکیب داده‌های تقریب زننده و خطا

ورودی. c,n,k,u,d,r,t,met همانند پارامترهای برنامهٔ decompdemo1 تعریف می‌شوند.

met یک نویسه است که نوع نرم بکار رفته در فرایند تجزیهٔ اعمال شده روی پارامترهای ورودی را تعیین می‌کند و می‌تواند یکی از نویسه‌های '2' برای نرم L_2 گسسته، 'm' برای نرم L_∞ گسسته و یا 's' برای نرم L_∞ پیوسته باشد.

خروجی. c, n, u مقادیر پارامتر بعد از پایان فرایند ترکیب هستند.

```
function [c, n, u]=compdemo(c,n,k,u,d,r,t,met)
% Demonstration of BSpline Composition:
%   [c, n, u]=compdemo(c,n,k,u,d,r,t,met)
%   met='2' -> l2, met='m' -> lmax, met='s' -> Lsup
while r
   [di mu]=ffknot(n,k,u,t(r),1,n+k);
   if met(1)=='2'
      [c n u]=reconstl2(c,n,k,u,t(r),di,mu,d(:,r));
   elseif met(1)=='m'
      [c n u]=reconstmax(c,n,k,u,t(r),di,mu,d(:,r));
   elseif met(1)=='s'
      [c n u]=reconstsup(c,n,k,u,t(r),di,mu,d(:,r));
   else
      return;
   end
   r=r-1;
end
```

کتاب‌نامه

[1] R. Barnhill and R. F. Riesenfeld, *Computer Aided Geometric Design*, Academic Press, 1974.

[2] P. Bézier, *Procédé de définition numérique des courbes et surfaces non mathématiques*, Automatisme, XIII 1968.

[3] P. Bézier, *Essay de définition numérique des courbes et surfaces expérimentales*, Ph.D. thesis, University of Paris VI, 1977.

[4] J. Blatter and L. Schumaker, *Continiuous selection and maximal alternators for spline approximation*, J. Approx. Theory, 38 (1983), pp. 71–80.

[5] E. W. Cheney, *Introduction to Approximation Theory*, McGraw-Hill, New York, 1966.

[6] C. Chui, *An Introduction to Wavelets*, Academic Press, Boston, 1992.

[7] S. Coons, *Surfaces for computer aided design*, Technical Report, MIT, Springfield, VA 22161, 1964. Available as AD 663 504 from the National Technical Information service.

[8] S. Coons, *Surfaces for computer aided design of space forms*, Technical Report, MIT, 1967. Project MAC-TR 41.

[9] M. Cox, *The numerical evaluation of B-splines*, DNAC 4, National Physical Laboratory, 1971.

[10] M. Dæhlen and T. Lyche, *Decomposition of spline*, in Mathematical Methods in Computer Aided Geometric Design II, T. Lyche and L. Schumaker, eds., Academic Press, 1992, pp. 135–160.

[11] C. de Boor, *On calculating with B-splines*, J. Approx. Theory, 6 (1972), pp. 50–62.

[12] C. de Boor, *On local linear functionals which vanish at all B-splines but one*, in Theory of approximation with applications, A. Law and B. Sahney, eds., Academic Press, 1976, pp. 120–145.

[13] C. de Boor, *A Practical Guide to Splines*, Springer, 1978.

[14] C. de Boor, *A geomertic proof for the variation diminishing property of B-spline approximation*, J. Approx. Theory, 37 (1983), pp. 5–11.

[15] P. de Casteljau, *Outillages méthods calcul*, Technical Report, A. Citroen, Paris, 1959.

[16] P. de Casteljau, *Courbes et surfaces à poles*, technical report, A. Citroen, Paris, 1963.

[17] W. Degen, *Best approximation of parametric curves by spline*, in Mathematical Methods in Computer Aided Geometric Design II, T. Lyche and L. Schumaker, eds., Academic Press, 1992, pp. 171–184.

[18] T. L. E. Cohen and R. Riesenfeld, *Discrete B-splines and subdivsion techniques in computer aided geometric design and computer graphics*, Comp. Graphics and Image Process., 14 (1980), pp. 87–111.

[19] M. Eck, *Degree reduction of Bézier curves*, Computer Aided Geometric Design, 10 (1993), pp. 237–251.

[20] M. Eck and J. Hadenfeld, *Knot removal for B-spline curves*, Computer Aided Geometric Design, 12 (1995), pp. 259–282.

[21] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design*, Acamedic Press, 3rd ed., 1993.

[22] A. Forrest, *Interactive interpolation and approximation by Bézier polynomials*, Computer J., 15 (1972), pp. 71–79.

[23] D. Handscomb, *Knot elimination: reversal of the oslo algorithm*, in ISNM, Vol. 81, Birkhäuser, 1987, pp. 103–111.

[24] C. L. Lawson and R. Hanson, *Solving Least Squares Problems*, Printice-Hall, Engelwood Cliffs, NJ, 1974.

[25] T. Lyche, *Knot removal for spline curves and surfaces*, in Approximation Theory VII, C. K. Chui and L. Schumaker, eds., Academic Press, 1992, pp. 207–226.

[26] T. Lyche and K. Mørken, *A data-reduction strategy for splines with applications to the approximation of functions and data*, IMA J. Numer. Anal., 8 (1988), pp. 185–208.

[27] T. Lyche and K. Mørken, *A discrete approach to knot removal and degree reduction algorithms for splines*, in Algorithms for Approximation, J. Mason and M. Cox, eds., Clarendon Press, Oxford, 1987, pp. 67–82.

[28] T. Lyche and K. Mørken, *Knot removal for parametric B-spline curves and surfaces*, Computer Aided Geometric Design, 4 (1987), pp. 217–230.

[29] S. Mallat, *Multiresolution approximations and wavelet orthonormal bases of* $L^2(\mathbb{R})$, Trans. Amer. Math. Soc., 315 (1989), pp. 69–87.

[30] G. Nürnberger, *Approximation by Spline Function*, Springer-Verlag, Berlin, 1989.

[31] G. Nürnberger and M. Sommer, *Alternation for best spline approximations*, Numer. Math., 41 (1983), pp. 207–221.

[32] G. Nürnberger and M. Sommer, *A remez type algorithm for spline functions*, Numer. Math., 41 (1983), pp. 117–146.

[33] R. H. Bartels and J. C. Beatty and B. A. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*, Morgan Kaufmann Publishers, 1987.

[34] R. Riesendfeld, *Applications of B-spline approximation to geometric problems of computer-aided design*, Ph.D. Thesis, Syracuse Univ., 1973.

[35] R. Schaback, *Error estimates for approximation from control nets*, Computer Aided Geometric Design, 10 (1993), pp. 57–66.

[36] I. Schoenberg, *Contributions to the problem of approximation of equidistant data by analytic functions*, Quart. Appl. Math, 4 (1946), pp. 45–99.

[37] L. Schumaker, *On shape-preserving quadratic spline interpolation*, SIAM J. Numer. Anal., 20 (1983), pp. 854–864.

[38] L. Schumaker and S. Stanley, *Shape-preserving knot removal*, Computer Aided Geometric Design, 13 (1996), pp. 851–872.

[39] W. Tiller, *Knot-removal algorithms for nurbs curves and surfaces*, Computer Aided Design, 24 (1992), pp. 445–453.

[40] M. von Golitschek and L. Schumaker, *Data fitting by penalized least squares*, in Algorithms for Approximation II, M. Cox and J. Mason, eds., Chapman & Hall, London, 1990, pp. 210–227.

[41] M. Watkins and A. Worsey, *Degree reduction for Bézier curves*, Computer Aided Design, 20 (1988), pp. 398–405.

[42] H. Wolters, *A method for computing best segment approximations*, TR 93-001, Arizona State University, 1993.

[۴۳] **فامیل سماواتی، فرامرز**: مدل‌سازیهای هندسی در کامپیوتر و کاربرد آنها در طراحی فونتها، پایان‌نامهٔ کارشناسی‌ارشد، دانشگاه صنعتی شریف، ۱۳۷۰.

[۴۴] **فامیل سماواتی، فرامرز**: تفکیک پذیری چندگانه در گرافیک کامپیوتری، پایان‌نامهٔ دکتری ریاضی، دانشگاه صنعتی شریف، فروردین ۱۳۷۸.

Knot removal for B-spline curves

Abstract

The purpose of knot removal for a B-spline curve is to obtain a new B-spline curve defined on the knot sequence from which some knots are removed out of an initial sequence, so that the curve has the least difference with the initial curve with respect to some norms.

In this thesis, the problem of removing one inner knot from the knot sequence of a B-spline curve is discussed. Doing so, a local geometric construction of the new control points is introduced based on two special inverses of knot insertion matrices. Then the free weight coefficients appearing in this general construction are determined by minimizing three norms of discrete L_∞ , L_2 and continiuous L_∞, between the old and new curves. Depending on each of the three norms, an approximation problem comes into consideration, the best of which is continiuous L_∞ approximation, calculated by a particular type of Remez algorithm.

By implementing some composition and decomposition algorithms of B-spline, based on the methods of removing a single knot, some applications can be considered such as the problem of knot ranking for joint removal of many knots as a special application, and also many practical applications like curve and surface fiiting, signal processing, image compression and approximation of functions out of spline space with free-knot sequence.

Keywords: *knot reomval, knot insertion, discrete L_∞ approximation, discrete L_2 approximation, continiuous L_∞ approximation, Remez algorithm, composition and decomposition of B-splines, nonuniform knot sequence, knot ranking, signal processing, good knot problem.*

Sharif University of Technology

Department of Mathematical Sciences

Master of Science Thesis in

Applied Mathematics

Title

Knot removal for B-spline curves

Supervisor

Professor N. Mahdavi-Amiri

Author

Ali Katanforush

September 1999