Convergence of corner cutting algorithms refining points and nets of functions

Lucia Romani

University of Milano-Bicocca, Italy

Joint work with:
Costanza Conti (University of Firenze, Italy)
Nira Dyn (Tel-Aviv University, Israel)

"Multivariate Approximation and Interpolation with Applications"

Luminy, September 19-23, 2016

Related literature:

☞ de Boor, C.: Cutting corners always works, CAGD (1987)

☞ Gregory, J.A., Qu, R.: Nonuniform corner cutting, CAGD (1996)

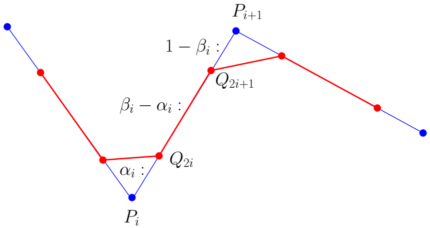
Goals of our work:

- Give a *very simple proof* of the fact that a corner cutting algorithm for points always converges if the corner cutting weights satisfy the conditions assumed by Gregory and Qu.
- Extend this result to the bivariate setting to show convergence of bivariate corner cutting algorithms refining nets of functions.

Preliminary definitions

Definition (Corner cutting weights)

$$\mathcal{W} = \left\{ (\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \ell(\mathbb{Z}) \times \ell(\mathbb{Z}) \; : \; \inf_{i \in \mathbb{Z}} \{\alpha_i,\, 1-\beta_i,\, \beta_i - \alpha_i\} > 0 \right\}$$



Examples: Chaikin weights

$$(\alpha_i, \beta_i) = \left(\frac{1}{4}, \frac{3}{4}\right)$$

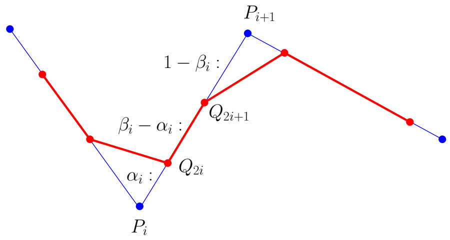
de Rham weights

$$(\alpha_i, \beta_i) = \left(\frac{1}{3}, \frac{2}{3}\right)$$

Definition ($CC_{(\boldsymbol{\alpha},\boldsymbol{\beta})}$-operator)

Let $(\boldsymbol{\alpha},\boldsymbol{\beta}) \in \mathcal{W}$ and $\mathbf{P} = (P_i)_{i\in\mathbb{Z}}$, $P_i \in \mathbb{R}^n$.
The *corner cutting* operator acting on $\mathbf{P} \in \ell^n(\mathbb{Z})$ is defined as

$$CC_{(\boldsymbol{\alpha},\boldsymbol{\beta})} \;:\; \ell^n(\mathbb{Z}) \longrightarrow \ell^n(\mathbb{Z}),$$
$$Q_{2i} := \left(CC_{(\boldsymbol{\alpha},\boldsymbol{\beta})}(\mathbf{P})\right)_{2i} = (1-\alpha_i)P_i + \alpha_i P_{i+1},$$
$$Q_{2i+1} := \left(CC_{(\boldsymbol{\alpha},\boldsymbol{\beta})}(\mathbf{P})\right)_{2i+1} = (1-\beta_i)P_i + \beta_i P_{i+1}.$$



$$\alpha_i = \frac{\|Q_{2i} - P_i\|_2}{\|P_{i+1} - P_i\|_2}$$

$$1-\beta_i = \frac{\|P_{i+1} - Q_{2i+1}\|_2}{\|P_{i+1} - P_i\|_2}$$

$$\beta_i-\alpha_i = \frac{\|Q_{2i+1} - Q_{2i}\|_2}{\|P_{i+1} - P_i\|_2}$$

The corner cutting algorithm

The CC-algorithm

Input: $\mathbf{P}^{[0]} \in \ell^n(\mathbb{Z})$

For $k = 0, 1, \dots,$

Input: $(\boldsymbol{\alpha}^{[k]}, \boldsymbol{\beta}^{[k]}) \in \mathcal{W}$

Compute $\mathbf{P}^{[k+1]} = CC_{(\boldsymbol{\alpha}^{[k]},\, \boldsymbol{\beta}^{[k]})}(\mathbf{P}^{[k]})$

The corner cutting algorithm

The CC-algorithm

Input: $\mathbf{P}^{[0]} \in \ell^n(\mathbb{Z})$

For $k = 0, 1, \ldots,$

Input: $(\boldsymbol{\alpha}^{[k]}, \boldsymbol{\beta}^{[k]}) \in \mathcal{W}$

Compute $\mathbf{P}^{[k+1]} = CC_{(\boldsymbol{\alpha}^{[k]},\, \boldsymbol{\beta}^{[k]})}(\mathbf{P}^{[k]})$

☞ The CC-algorithm is applied to the n scalar sequences obtained from the components of $\mathbf{P}^{[0]}$.

The corner cutting algorithm

The CC-algorithm

Input: $\mathbf{P}^{[0]} \in \ell^n(\mathbb{Z})$

For $k = 0, 1, \ldots,$

Input: $(\boldsymbol{\alpha}^{[k]}, \boldsymbol{\beta}^{[k]}) \in \mathcal{W}$

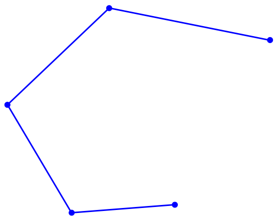
Compute $\mathbf{P}^{[k+1]} = CC_{(\boldsymbol{\alpha}^{[k]}, \boldsymbol{\beta}^{[k]})}(\mathbf{P}^{[k]})$

☞ The CC-algorithm is applied to the n scalar sequences obtained from the components of $\mathbf{P}^{[0]}$.

Example ($n = 2$):

$$\alpha_i^{[k]} = \begin{cases} \frac{1}{2(1+\cos(2^{-(k+2)\pi}))}, & \text{if } i \text{ even} \\ \frac{1}{2(1+\cosh(2^{-(k+1)}))}, & \text{if } i \text{ odd} \end{cases}$$

$$\beta_i^{[k]} = \begin{cases} 1 - \frac{1}{2(1+\cos(2^{-(k+2)\pi}))}, & \text{if } i \text{ even} \\ 1 - \frac{1}{2(1+\cosh(2^{-(k+1)}))}, & \text{if } i \text{ odd} \end{cases}$$



$\mathbf{P}^{[0]}$

The corner cutting algorithm

The CC-algorithm

Input: $\mathbf{P}^{[0]} \in \ell^n(\mathbb{Z})$

For $k = 0, 1, \ldots,$

Input: $(\boldsymbol{\alpha}^{[k]}, \boldsymbol{\beta}^{[k]}) \in \mathcal{W}$

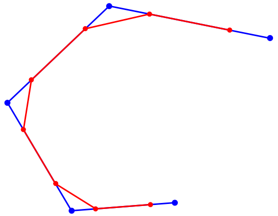
Compute $\mathbf{P}^{[k+1]} = CC_{(\boldsymbol{\alpha}^{[k]}, \boldsymbol{\beta}^{[k]})}(\mathbf{P}^{[k]})$

☞ The CC-algorithm is applied to the n scalar sequences obtained from the components of $\mathbf{P}^{[0]}$.

Example ($n = 2$):

$$\alpha_i^{[k]} = \begin{cases} \frac{1}{2(1+\cos(2^{-(k+2)\pi}))}, & \text{if } i \text{ even} \\ \frac{1}{2(1+\cosh(2^{-(k+1)}))}, & \text{if } i \text{ odd} \end{cases}$$

$$\beta_i^{[k]} = \begin{cases} 1 - \frac{1}{2(1+\cos(2^{-(k+2)\pi}))}, & \text{if } i \text{ even} \\ 1 - \frac{1}{2(1+\cosh(2^{-(k+1)}))}, & \text{if } i \text{ odd} \end{cases}$$



$\mathbf{P}^{[1]}$

The corner cutting algorithm

The CC-algorithm

Input: $\mathbf{P}^{[0]} \in \ell^n(\mathbb{Z})$

For $k = 0, 1, \ldots,$

Input: $(\boldsymbol{\alpha}^{[k]}, \boldsymbol{\beta}^{[k]}) \in \mathcal{W}$

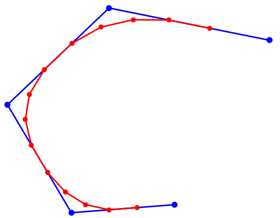
Compute $\mathbf{P}^{[k+1]} = CC_{(\boldsymbol{\alpha}^{[k]}, \boldsymbol{\beta}^{[k]})}(\mathbf{P}^{[k]})$

☞ The CC-algorithm is applied to the n scalar sequences obtained from the components of $\mathbf{P}^{[0]}$.

Example ($n = 2$):

$$\alpha_i^{[k]} = \begin{cases} \frac{1}{2(1+\cos(2^{-(k+2)\pi}))}, & \text{if } i \text{ even} \\ \frac{1}{2(1+\cosh(2^{-(k+1)}))}, & \text{if } i \text{ odd} \end{cases}$$

$$\beta_i^{[k]} = \begin{cases} 1 - \frac{1}{2(1+\cos(2^{-(k+2)\pi}))}, & \text{if } i \text{ even} \\ 1 - \frac{1}{2(1+\cosh(2^{-(k+1)}))}, & \text{if } i \text{ odd} \end{cases}$$



$\mathbf{P}^{[2]}$

The corner cutting algorithm

The CC-algorithm

Input: $\mathbf{P}^{[0]} \in \ell^n(\mathbb{Z})$

For $k = 0, 1, \ldots,$

Input: $(\boldsymbol{\alpha}^{[k]}, \boldsymbol{\beta}^{[k]}) \in \mathcal{W}$

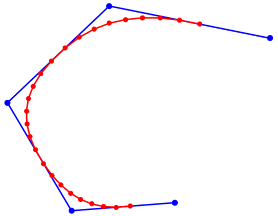
Compute $\mathbf{P}^{[k+1]} = CC_{(\boldsymbol{\alpha}^{[k]}, \boldsymbol{\beta}^{[k]})}(\mathbf{P}^{[k]})$

☞ The CC-algorithm is applied to the n scalar sequences obtained from the components of $\mathbf{P}^{[0]}$.

Example ($n = 2$):

$$\alpha_i^{[k]} = \begin{cases} \frac{1}{2(1+\cos(2^{-(k+2)\pi}))}, & \text{if } i \text{ even} \\ \frac{1}{2(1+\cosh(2^{-(k+1)}))}, & \text{if } i \text{ odd} \end{cases}$$

$$\beta_i^{[k]} = \begin{cases} 1 - \frac{1}{2(1+\cos(2^{-(k+2)\pi}))}, & \text{if } i \text{ even} \\ 1 - \frac{1}{2(1+\cosh(2^{-(k+1)}))}, & \text{if } i \text{ odd} \end{cases}$$



$\mathbf{P}^{[3]}$

The corner cutting algorithm

The CC-algorithm

Input: $\mathbf{P}^{[0]} \in \ell^n(\mathbb{Z})$

For $k = 0, 1, \ldots,$

Input: $(\boldsymbol{\alpha}^{[k]}, \boldsymbol{\beta}^{[k]}) \in \mathcal{W}$

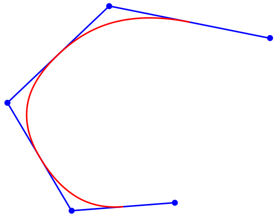
Compute $\mathbf{P}^{[k+1]} = CC_{(\boldsymbol{\alpha}^{[k]}, \boldsymbol{\beta}^{[k]})}(\mathbf{P}^{[k]})$

☞ The CC-algorithm is applied to the n scalar sequences obtained from the components of $\mathbf{P}^{[0]}$.

Example ($n = 2$):

$$\alpha_i^{[k]} = \begin{cases} \frac{1}{2(1+\cos(2^{-(k+2)\pi}))}, & \text{if } i \text{ even} \\ \frac{1}{2(1+\cosh(2^{-(k+1)}))}, & \text{if } i \text{ odd} \end{cases}$$

$$\beta_i^{[k]} = \begin{cases} 1 - \frac{1}{2(1+\cos(2^{-(k+2)\pi}))}, & \text{if } i \text{ even} \\ 1 - \frac{1}{2(1+\cosh(2^{-(k+1)}))}, & \text{if } i \text{ odd} \end{cases}$$



$$\lim_{k\to\infty} \mathbf{P}^{[k]}$$

Theorem I (Analysis of convergence)

For $\{(\boldsymbol{\alpha}^{[k]}, \boldsymbol{\beta}^{[k]}) \in \mathcal{W},\ k \geq 0\}$ such that

$$\mu = \sup_{k \geq 0} \mu^{[k]} < 1 \qquad \text{with} \qquad \mu^{[k]} = \sup_{i \in \mathbb{Z}} \{\beta_i^{[k]} - \alpha_i^{[k]},\ 1 - \beta_{i-1}^{[k]} + \alpha_i^{[k]}\}$$

the CC-algorithm converges for all $\mathbf{P}^{[0]} = \{P_i^{[0]} \in \mathbb{R}^n,\ i \in \mathbb{Z}\} \in \ell^n(\mathbb{Z})$ satisfying for a certain $L > 0$

$$\|P_{i+1}^{[0]} - P_i^{[0]}\|_\infty < L, \quad \forall i \in \mathbb{Z}.$$

Theorem I (Analysis of convergence)

For $\{(\boldsymbol{\alpha}^{[k]}, \boldsymbol{\beta}^{[k]}) \in \mathcal{W},\ k \geq 0\}$ such that

$$\boxed{\mu = \sup_{k\geq 0} \mu^{[k]} < 1 \qquad \text{with} \qquad \mu^{[k]} = \sup_{i\in\mathbb{Z}} \{\beta_i^{[k]} - \alpha_i^{[k]},\ 1 - \beta_{i-1}^{[k]} + \alpha_i^{[k]}\}}$$

the CC-algorithm converges for all $\mathbf{P}^{[0]} = \{P_i^{[0]} \in \mathbb{R}^n,\ i \in \mathbb{Z}\} \in \ell^n(\mathbb{Z})$ satisfying for a certain $L > 0$

$$\|P_{i+1}^{[0]} - P_i^{[0]}\|_\infty < L, \quad \forall i \in \mathbb{Z}. \qquad (\star)$$

☞ The assumption $(\star)$ is equivalent to requiring that the piecewise linear interpolant to the data $(i, P_i^{[0]})$, $i \in \mathbb{Z}$ is Lipschitz continuous (LipC) in $\mathbb{R}$ with Lipschitz constant L.

Theorem I (Analysis of convergence)

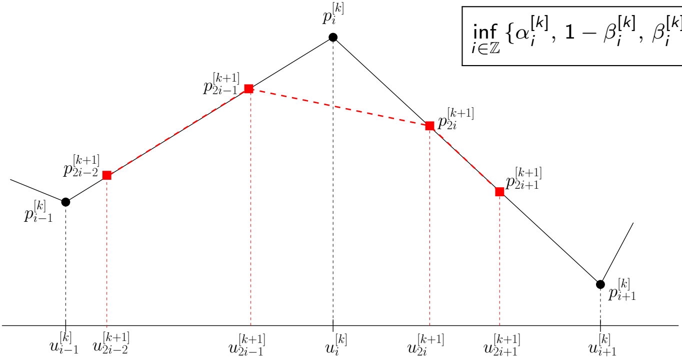
For $\{(\boldsymbol{\alpha}^{[k]}, \boldsymbol{\beta}^{[k]}) \in \mathcal{W},\ k \geq 0\}$ such that

$$\mu = \sup_{k \geq 0} \mu^{[k]} < 1 \qquad \text{with} \qquad \mu^{[k]} = \sup_{i \in \mathbb{Z}} \{\beta_i^{[k]} - \alpha_i^{[k]},\ 1 - \beta_{i-1}^{[k]} + \alpha_i^{[k]}\}$$

the CC-algorithm converges for all $\mathbf{P}^{[0]} = \{P_i^{[0]} \in \mathbb{R}^n, i \in \mathbb{Z}\} \in \ell^n(\mathbb{Z})$ satisfying for a certain $L > 0$

$$\|P_{i+1}^{[0]} - P_i^{[0]}\|_\infty < L, \quad \forall i \in \mathbb{Z}.$$

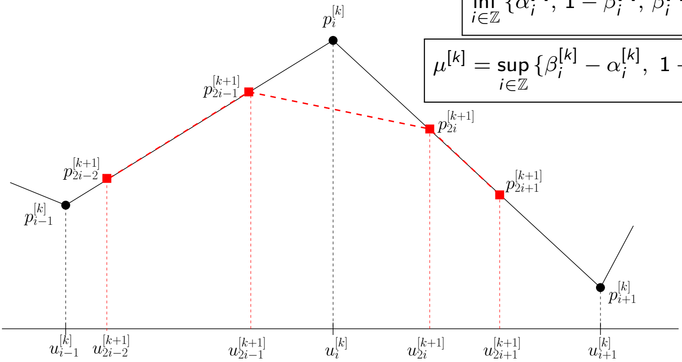
- $p_i^{[k]} \in \mathbb{R}$ one component of $P_i^{[k]} \in \mathbb{R}^n$
- $\mathcal{L}_k(\mathbf{p}^{[k]}) : \mathbb{R} \to \mathbb{R}$ piecewise linear interpolant to $(u_i^{[k]}, p_i^{[k]}),\ i \in \mathbb{Z}$
- $\mathbf{u}^{[k]}$: scalar sequence obtained from $\mathbf{u}^{[0]} = \mathbb{Z}$ after k steps of CC-algorithm.



$$\inf_{i\in\mathbb{Z}} \{\alpha_i^{[k]},\, 1-\beta_i^{[k]},\, \beta_i^{[k]}-\alpha_i^{[k]}\} > 0$$

$$u_{2i}^{[k]} = (1-\alpha_i^{[k-1]})u_i^{[k-1]} + \alpha_i^{[k-1]} u_{i+1}^{[k-1]}, \qquad u_{2i+1}^{[k]} = (1-\beta_i^{[k-1]})u_i^{[k-1]} + \beta_i^{[k-1]} u_{i+1}^{[k-1]}$$

- $u_{2i-1}^{[k+1]} - u_{2i-2}^{[k+1]} = (\beta_{i-1}^{[k]} - \alpha_{i-1}^{[k]})(u_i^{[k]} - u_{i-1}^{[k]})$
- $u_{2i}^{[k+1]} - u_{2i-1}^{[k+1]} = (1-\beta_{i-1}^{[k]})(u_i^{[k]} - u_{i-1}^{[k]}) + \alpha_i^{[k]}(u_{i+1}^{[k]} - u_i^{[k]})$

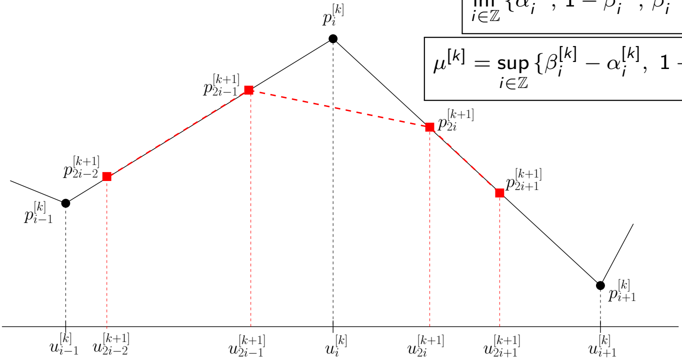


$$\inf_{i\in\mathbb{Z}} \{\alpha_i^{[k]},\ 1-\beta_i^{[k]},\ \beta_i^{[k]}-\alpha_i^{[k]}\} > 0$$

$$\mu^{[k]} = \sup_{i\in\mathbb{Z}} \{\beta_i^{[k]}-\alpha_i^{[k]},\ 1-\beta_{i-1}^{[k]}+\alpha_i^{[k]}\} < 1$$

- $u_{2i-1}^{[k+1]} - u_{2i-2}^{[k+1]} = (\beta_{i-1}^{[k]} - \alpha_{i-1}^{[k]})(u_i^{[k]} - u_{i-1}^{[k]})$
- $u_{2i}^{[k+1]} - u_{2i-1}^{[k+1]} = (1-\beta_{i-1}^{[k]})(u_i^{[k]} - u_{i-1}^{[k]}) + \alpha_i^{[k]}(u_{i+1}^{[k]} - u_i^{[k]})$

$$d^{[k+1]} := \sup_{i\in\mathbb{Z}}\{u_{2i+1}^{[k+1]} - u_{2i}^{[k+1]}\}$$



$$\inf_{i\in\mathbb{Z}} \{\alpha_i^{[k]},\ 1-\beta_i^{[k]},\ \beta_i^{[k]}-\alpha_i^{[k]}\} > 0$$

$$\mu^{[k]} = \sup_{i\in\mathbb{Z}} \{\beta_i^{[k]}-\alpha_i^{[k]},\ 1-\beta_{i-1}^{[k]}+\alpha_i^{[k]}\} < 1$$

- $u_{2i-1}^{[k+1]} - u_{2i-2}^{[k+1]} = (\beta_{i-1}^{[k]} - \alpha_{i-1}^{[k]})(u_i^{[k]} - u_{i-1}^{[k]})$
- $u_{2i}^{[k+1]} - u_{2i-1}^{[k+1]} = (1-\beta_{i-1}^{[k]})(u_i^{[k]} - u_{i-1}^{[k]}) + \alpha_i^{[k]}(u_{i+1}^{[k]} - u_i^{[k]})$

$$d^{[k+1]} := \sup_{i\in\mathbb{Z}}\{u_{2i+1}^{[k+1]} - u_{2i}^{[k+1]}\}$$

$$d^{[k+1]} \le \mu^{[k]}\, d^{[k]} \quad \Rightarrow \quad d^{[k+1]} < d^{[0]}\mu^{k+1} \quad \text{with} \quad \mu := \sup_{k\ge 0} \mu^{[k]} < 1$$

The proof consists in showing that $\{\mathcal{L}_k(\mathbf{p}^{[k]})\}_{k\geq 0}$ is a Cauchy sequence.

Key steps of the proof:

1. By assumption, $\mathcal{L}_0(\mathbf{p}^{[0]})$ is LipC in $\mathbb{R}$ with Lipschitz constant L.

The proof consists in showing that $\{\mathcal{L}_k(\mathbf{p}^{[k]})\}_{k\geq 0}$ is a Cauchy sequence.

Key steps of the proof:

1. By assumption, $\mathcal{L}_0(\mathbf{p}^{[0]})$ is LipC in $\mathbb{R}$ with Lipschitz constant L.
2. $\forall\, k \geq 0$ all points of $\mathbf{p}^{[k+1]}$ lie on $\mathcal{L}_k(\mathbf{p}^{[k]})$ thus, by the choice of $\mathbf{u}^{[k+1]}$, $|p_{i+1}^{[k+1]} - p_i^{[k]}| \leq L\, |u_{i+1}^{[k+1]} - u_i^{[k]}| \;\forall i \;\Rightarrow\; \mathcal{L}_k(\mathbf{p}^{[k]})$ LipC in $\mathbb{R}$ with constant L.

The proof consists in showing that $\{\mathcal{L}_k(\mathbf{p}^{[k]})\}_{k\geq 0}$ is a Cauchy sequence.

Key steps of the proof:

1. By assumption, $\mathcal{L}_0(\mathbf{p}^{[0]})$ is LipC in $\mathbb{R}$ with Lipschitz constant L.
2. $\forall\, k \geq 0$ all points of $\mathbf{p}^{[k+1]}$ lie on $\mathcal{L}_k(\mathbf{p}^{[k]})$ thus, by the choice of $\mathbf{u}^{[k+1]}$, $|p_{i+1}^{[k+1]} - p_i^{[k]}| \leq L\,|u_{i+1}^{[k+1]} - u_i^{[k]}|\ \forall i \ \Rightarrow\ \mathcal{L}_k(\mathbf{p}^{[k]})$ LipC in $\mathbb{R}$ with constant L.

Proposition (A)

If f is LipC on $[a,b]$ with Lipschitz constant L, then $p_1(x) := \frac{x-a}{b-a}f(b) + \frac{b-x}{b-a}f(a)$ satisfies

$$|p_1(x) - f(x)| \leq \frac{1}{2}(b-a)L, \quad \forall x \in [a,b]\,.$$

The proof consists in showing that $\{\mathcal{L}_k(\mathbf{p}^{[k]})\}_{k\geq 0}$ is a Cauchy sequence.

Key steps of the proof:

1. By assumption, $\mathcal{L}_0(\mathbf{p}^{[0]})$ is LipC in $\mathbb{R}$ with Lipschitz constant L.
2. $\forall\, k \geq 0$ all points of $\mathbf{p}^{[k+1]}$ lie on $\mathcal{L}_k(\mathbf{p}^{[k]})$ thus, by the choice of $\mathbf{u}^{[k+1]}$, $|p_{i+1}^{[k+1]} - p_i^{[k]}| \leq L\,|u_{i+1}^{[k+1]} - u_i^{[k]}|\ \forall i \ \Rightarrow\ \mathcal{L}_k(\mathbf{p}^{[k]})$ LipC in $\mathbb{R}$ with constant L.

Proposition (A)

If f is LipC on $[a,b]$ with Lipschitz constant L, then $p_1(x) := \frac{x-a}{b-a} f(b) + \frac{b-x}{b-a} f(a)$ satisfies

$$|p_1(x) - f(x)| \leq \frac{1}{2}(b-a)L, \quad \forall x \in [a,b]\,.$$

3. Since $\mathcal{L}_{k+1}(\mathbf{p}^{[k+1]})$ can be regarded as an approximation of the LipC function $\mathcal{L}_k(\mathbf{p}^{[k]})$, in view of Prop.(A)

$$|\mathcal{L}_{k+1}(\mathbf{p}^{[k+1]})(u) - \mathcal{L}_k(\mathbf{p}^{[k]})(u)| \leq \frac{1}{2}\,L\,d^{[k+1]} < \frac{1}{2}L\,d^{[0]}\,\mu^{k+1}$$

The proof consists in showing that $\{\mathcal{L}_k(\mathbf{p}^{[k]})\}_{k\geq 0}$ is a Cauchy sequence.

Key steps of the proof:

1. By assumption, $\mathcal{L}_0(\mathbf{p}^{[0]})$ is LipC in $\mathbb{R}$ with Lipschitz constant L.
2. $\forall\, k \geq 0$ all points of $\mathbf{p}^{[k+1]}$ lie on $\mathcal{L}_k(\mathbf{p}^{[k]})$ thus, by the choice of $\mathbf{u}^{[k+1]}$, $|p_{i+1}^{[k+1]} - p_i^{[k]}| \leq L\,|u_{i+1}^{[k+1]} - u_i^{[k]}|\ \forall i \ \Rightarrow\ \mathcal{L}_k(\mathbf{p}^{[k]})$ LipC in $\mathbb{R}$ with constant L.

Proposition (A)

If f is LipC on $[a,b]$ with Lipschitz constant L, then $p_1(x) := \frac{x-a}{b-a}f(b) + \frac{b-x}{b-a}f(a)$ satisfies

$$|p_1(x) - f(x)| \leq \frac{1}{2}(b-a)L, \quad \forall x \in [a,b]\,.$$

3. Since $\mathcal{L}_{k+1}(\mathbf{p}^{[k+1]})$ can be regarded as an approximation of the LipC function $\mathcal{L}_k(\mathbf{p}^{[k]})$, in view of Prop.(A)

$$|\mathcal{L}_{k+1}(\mathbf{p}^{[k+1]})(u) - \mathcal{L}_k(\mathbf{p}^{[k]})(u)| \leq \frac{1}{2}\,L\,d^{[k+1]} < \frac{1}{2}L\,d^{[0]}\,\mu^{k+1}$$

$$\Downarrow$$

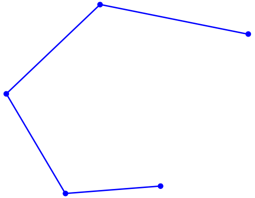
$$|\mathcal{L}_{k+r}(\mathbf{p}^{[k+r]})(u) - \mathcal{L}_k(\mathbf{p}^{[k]})(u)| \ \leq\ \frac{1}{2}L\,d^{[0]}\mu^{k+1}\left(\sum_{\ell=0}^{r-1}\mu^{\ell}\right). \qquad \square$$

The Generalized Lane-Riesenfeld algorithm

Lane-Riesenfeld: at each refinement level Chaikin corner cutting is followed by m averaging steps ☞ degree-$(m+2)$ splines in the limit

The Generalized Lane-Riesenfeld algorithm

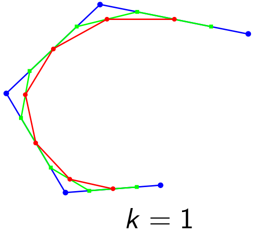
Lane-Riesenfeld: at each refinement level Chaikin corner cutting is followed by m averaging steps ☞ degree-$(m+2)$ splines in the limit



$k = 0$

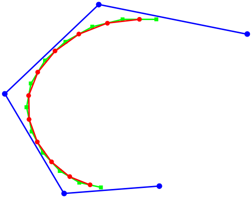
The Generalized Lane-Riesenfeld algorithm

Lane-Riesenfeld: at each refinement level Chaikin corner cutting is followed by m averaging steps ☞ degree-$(m+2)$ splines in the limit



The Generalized Lane-Riesenfeld algorithm

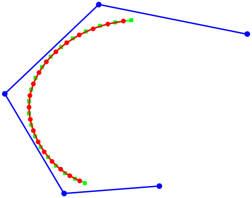
Lane-Riesenfeld: at each refinement level Chaikin corner cutting is followed by m averaging steps ☞ degree-$(m+2)$ splines in the limit



$k = 2$

The Generalized Lane-Riesenfeld algorithm

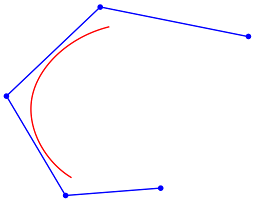
Lane-Riesenfeld: at each refinement level Chaikin corner cutting is followed by m averaging steps ☞ degree-$(m+2)$ splines in the limit



$k = 3$

The Generalized Lane-Riesenfeld algorithm

Lane-Riesenfeld: at each refinement level Chaikin corner cutting is followed by m averaging steps ☞ degree-$(m+2)$ splines in the limit



$k \to \infty$

The Generalized Lane-Riesenfeld algorithm

The GLR-algorithm

Input: $\mathbf{P}^{[0]} \in \ell^n(\mathbb{Z})$

For $k = 0, 1, \ldots$

$\mathbf{P}^{[k+1,0]} = CC_{(\boldsymbol{\alpha}^{[k]},\, \boldsymbol{\beta}^{[k]})}(\mathbf{P}^{[k]})$ with $(\boldsymbol{\alpha}^{[k]}, \boldsymbol{\beta}^{[k]}) \in \mathcal{W}$

Input: $\mathbf{w}^{[k]} \in \ell(\mathbb{Z})$ with $0 < w_i^{[k]} < 1 \quad \forall i \in \mathbb{Z},\ k \geq 0$

For $j = 0, \ldots, m_k - 1$ $(m_k \in \mathbb{N}_0$ s.t. $m_k < M \quad \forall k \geq 0)$

$\mathbf{P}^{[k+1,j+1]} = A^{[k]}\mathbf{P}^{[k+1,j]}$ with $(A^{[k]}\mathbf{P})_i = (1 - w_i^{[k]})P_i + w_i^{[k]}P_{i+1}$

$\mathbf{P}^{[k+1]} = \mathbf{P}^{[k+1,m_k]}$

The Generalized Lane-Riesenfeld algorithm

The GLR-algorithm

Input: $\mathbf{P}^{[0]} \in \ell^n(\mathbb{Z})$

For $k = 0, 1, \ldots$

$$\mathbf{P}^{[k+1,0]} = CC_{(\boldsymbol{\alpha}^{[k]},\, \boldsymbol{\beta}^{[k]})}(\mathbf{P}^{[k]}) \quad \text{with} \quad (\boldsymbol{\alpha}^{[k]}, \boldsymbol{\beta}^{[k]}) \in \mathcal{W}$$

Input: $\mathbf{w}^{[k]} \in \ell(\mathbb{Z})$ with $0 < w_i^{[k]} < 1 \quad \forall i \in \mathbb{Z},\ k \geq 0$

For $j = 0, \ldots, m_k - 1$ $\ (m_k \in \mathbb{N}_0$ s.t. $m_k < M \quad \forall k \geq 0)$

$$\mathbf{P}^{[k+1,j+1]} = A^{[k]}\mathbf{P}^{[k+1,j]} \quad \text{with} \quad (A^{[k]}\mathbf{P})_i = (1 - w_i^{[k]})P_i + w_i^{[k]}P_{i+1}$$

$$\mathbf{P}^{[k+1]} = \mathbf{P}^{[k+1,m_k]}$$

☞ Convergence is still guaranteed by the fact that

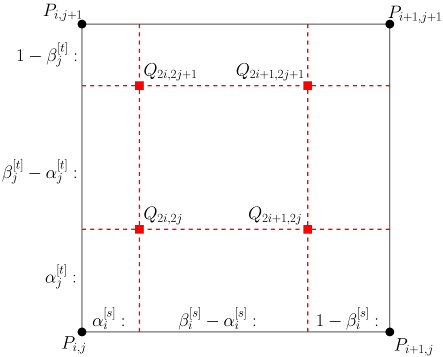
$$|\mathcal{L}_{k+1}(\mathbf{p}^{[k+1]})(u) - \mathcal{L}_k(\mathbf{p}^{[k]})(u)| < \tfrac{1}{2}\, ML\, d^{[k+1]}.$$

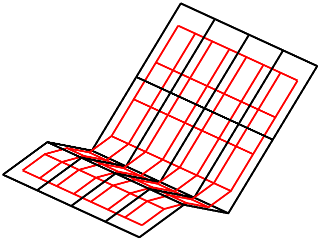
Generalization to nets of functions

From point corner cutting to net corner cutting

☞ **Point corner cutting (2D case)**:

For a given $\mathbf{P}^{[k]} \in \ell^n(\mathbb{Z}^2)$, we define $\mathbf{P}^{[k+1]}$ by sampling the *piecewise bilinear interpolant* to $\mathbf{P}^{[k]}$ at the values of s and t specified by $(\boldsymbol{\alpha}^{[s],[k]}, \boldsymbol{\beta}^{[s],[k]})$ and $(\boldsymbol{\alpha}^{[t],[k]}, \boldsymbol{\beta}^{[t],[k]})$.





From point corner cutting to net corner cutting

☞ **Point corner cutting (2D case)**:

For a given $\mathbf{P}^{[k]} \in \ell^n(\mathbb{Z}^2)$, we define $\mathbf{P}^{[k+1]}$ by sampling the *piecewise bilinear interpolant* to $\mathbf{P}^{[k]}$ at the values of s and t specified by $(\boldsymbol{\alpha}^{[s],[k]}, \boldsymbol{\beta}^{[s],[k]})$ and $(\boldsymbol{\alpha}^{[t],[k]}, \boldsymbol{\beta}^{[t],[k]})$.

☞ **Net corner cutting (rough idea)**:

For a given $N^{[k]}$, we define $N^{[k+1]}$ by sampling the *piecewise Coons interpolant* to $N^{[k]}$ at the values of s and t specified by $(\boldsymbol{\alpha}^{[s],[k]}, \boldsymbol{\beta}^{[s],[k]})$ and $(\boldsymbol{\alpha}^{[t],[k]}, \boldsymbol{\beta}^{[t],[k]})$.

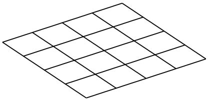
Grid of lines and net of functions

We consider a net of functions $N(T)$ defined on a *grid of lines*

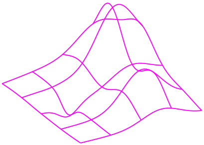
$$T \equiv T\Big(\big(\underbrace{\mathbf{h}^{[s]}}_{h_\ell^{[s]}:=s_{\ell+1}-s_\ell,\,\ell\in\mathbb{Z}},\mathbf{h}^{[t]}\big),\underbrace{O}_{(x_0,y_0)}\Big) := \{\mathbb{R}\times t_i,\ i\in\mathbb{Z}\}\cup\{s_j\times\mathbb{R},\ j\in\mathbb{Z}\}$$

$N := N(T)$ is a continuous bivariate function defined on the *grid of lines* T, which consists of the following continuous univariate functions:

$\{N(s,t_i)\}_{,\, i\in\mathbb{Z}}$
$\{N(s_j,t)\}_{,\, j\in\mathbb{Z}}$ (called the u-functions)



T

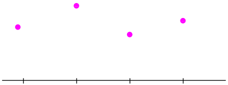


$N(T)$

Analogous to polyline definition

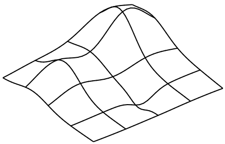
Point abscissae: $i\in\mathbb{Z}$

Point values: $\{f(i)\}_{i\in\mathbb{Z}}$



Definition (Compatible net of functions)

Let $\phi_i(s) := \{N(s, t_i)\}_{i\in\mathbb{Z}}$ and $\psi_j(t) := \{N(s_j, t)\}_{j\in\mathbb{Z}}$.
A net of functions N is said to be *compatible* if $\phi_i(s_j) = \psi_j(t_i)$ $\forall i, j \in \mathbb{Z}$.



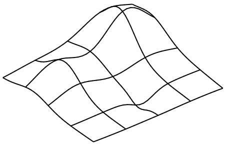
Compatible net

Definition (Compatible net of functions)

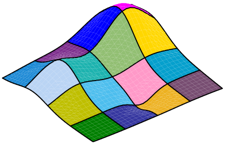
Let $\phi_i(s) := \{N(s, t_i)\}_{i\in\mathbb{Z}}$ and $\psi_j(t) := \{N(s_j, t)\}_{j\in\mathbb{Z}}$.
A net of functions N is said to be *compatible* if $\phi_i(s_j) = \psi_j(t_i)$ $\forall i, j \in \mathbb{Z}$.

Definition (Piecewise Coons patch)

We denote by $\mathcal{C}(N)$ the *piecewise Coons patch* interpolating a compatible net N.



Compatible net



Piecewise Coons patch

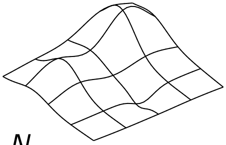
Definition (The operator $BC_{(\boldsymbol{\alpha},\,\boldsymbol{\beta})}(\mathcal{C}(N))$)

We denote by $BC_{(\boldsymbol{\alpha},\,\boldsymbol{\beta})}(\mathcal{C}(N))$ the net of u-functions obtained by sampling the piecewise Coons patch $\mathcal{C}(N)$ at the values of s and t specified by $(\boldsymbol{\alpha}^{[s]},\ \boldsymbol{\beta}^{[s]}) \in \mathcal{W}$ and $(\boldsymbol{\alpha}^{[t]},\ \boldsymbol{\beta}^{[t]}) \in \mathcal{W}$, i.e.,

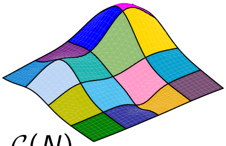
$$\begin{aligned} BC_{(\boldsymbol{\alpha},\,\boldsymbol{\beta})}(\mathcal{C}(N)) \;:=\; & \{\mathcal{C}(N)(s, t_j + \alpha_j^{[t]} h_j^{[t]}),\ \ \mathcal{C}(N)(s, t_j + \beta_j^{[t]} h_j^{[t]}),\ j \in \mathbb{Z}\} \\ \cup\ & \{\mathcal{C}(N)(s_i + \alpha_i^{[s]} h_i^{[s]}, t),\ \ \mathcal{C}(N)(s_i + \beta_i^{[s]} h_i^{[s]}, t),\ i \in \mathbb{Z}\} \end{aligned}$$

where

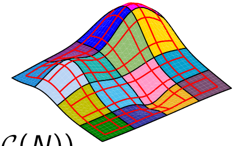
$(s,t) \in [s_i, s_{i+1}] \times [t_j, t_{j+1}]$ and $h_i^{[s]} = s_{i+1} - s_i,\ h_j^{[t]} = t_{j+1} - t_j,\ i, j \in \mathbb{Z}$.



N



$\mathcal{C}(N)$



$BC_{(\boldsymbol{\alpha},\,\boldsymbol{\beta})}(\mathcal{C}(N))$

The corner cutting algorithm for nets of functions

The CC-algorithm for nets

Input: a compatible net $N^{[0]}$

For $k = 0, 1, \ldots$

Input: $(\boldsymbol{\alpha}^{[s],[k]}, \boldsymbol{\beta}^{[s],[k]}) \in \mathcal{W}$

and $(\boldsymbol{\alpha}^{[t],[k]}, \boldsymbol{\beta}^{[t],[k]}) \in \mathcal{W}$

Compute

$N^{[k+1]} := BC_{(\boldsymbol{\alpha}^{[k]}, \boldsymbol{\beta}^{[k]})}(\mathcal{C}(N^{[k]}))$

The corner cutting algorithm for nets of functions

The CC-algorithm for nets

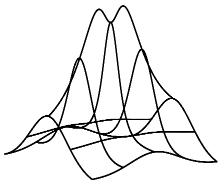
Input: a compatible net $N^{[0]}$

For $k = 0, 1, \ldots$

Input: $(\boldsymbol{\alpha}^{[s],[k]}, \boldsymbol{\beta}^{[s],[k]}) \in \mathcal{W}$
and $(\boldsymbol{\alpha}^{[t],[k]}, \boldsymbol{\beta}^{[t],[k]}) \in \mathcal{W}$

Compute
$N^{[k+1]} := BC_{(\boldsymbol{\alpha}^{[k]}, \boldsymbol{\beta}^{[k]})}(\mathcal{C}(N^{[k]}))$

Example:



$N^{[0]}$

The corner cutting algorithm for nets of functions

The CC-algorithm for nets

Input: a compatible net $N^{[0]}$

For $k = 0, 1, \ldots$

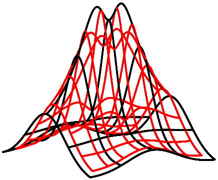
Input: $(\boldsymbol{\alpha}^{[s],[k]}, \boldsymbol{\beta}^{[s],[k]}) \in \mathcal{W}$

and $(\boldsymbol{\alpha}^{[t],[k]}, \boldsymbol{\beta}^{[t],[k]}) \in \mathcal{W}$

Compute

$N^{[k+1]} := BC_{(\boldsymbol{\alpha}^{[k]}, \boldsymbol{\beta}^{[k]})}(\mathcal{C}(N^{[k]}))$

Example:



$N^{[1]}$

The corner cutting algorithm for nets of functions

The CC-algorithm for nets

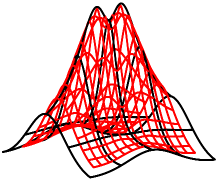
Input: a compatible net $N^{[0]}$

For $k = 0, 1, \ldots$

Input: $(\boldsymbol{\alpha}^{[s],[k]}, \boldsymbol{\beta}^{[s],[k]}) \in \mathcal{W}$
and $(\boldsymbol{\alpha}^{[t],[k]}, \boldsymbol{\beta}^{[t],[k]}) \in \mathcal{W}$

Compute
$N^{[k+1]} := BC_{(\boldsymbol{\alpha}^{[k]}, \boldsymbol{\beta}^{[k]})}(\mathcal{C}(N^{[k]}))$

Example:



$N^{[2]}$

The corner cutting algorithm for nets of functions

The CC-algorithm for nets

<u>Input</u>: a compatible net $N^{[0]}$

For $k = 0, 1, \ldots$

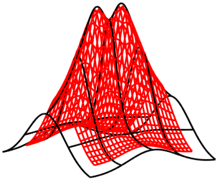
Input: $(\boldsymbol{\alpha}^{[s],[k]}, \boldsymbol{\beta}^{[s],[k]}) \in \mathcal{W}$

and $(\boldsymbol{\alpha}^{[t],[k]}, \boldsymbol{\beta}^{[t],[k]}) \in \mathcal{W}$

Compute

$N^{[k+1]} := BC_{(\boldsymbol{\alpha}^{[k]}, \boldsymbol{\beta}^{[k]})}(\mathcal{C}(N^{[k]}))$

Example:



$N^{[3]}$

The corner cutting algorithm for nets of functions

The CC-algorithm for nets

Input: a compatible net $N^{[0]}$

For $k = 0, 1, \ldots$

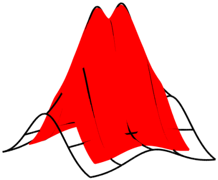
Input: $(\alpha^{[s],[k]}, \beta^{[s],[k]}) \in \mathcal{W}$

and $(\alpha^{[t],[k]}, \beta^{[t],[k]}) \in \mathcal{W}$

Compute

$N^{[k+1]} := BC_{(\alpha^{[k]},\, \beta^{[k]})}(\mathcal{C}(N^{[k]}))$

Example:



$N^{[4]}$

The corner cutting algorithm for nets of functions

The CC-algorithm for nets

Input: a compatible net $N^{[0]}$

For $k = 0, 1, \ldots$

Input: $(\alpha^{[s],[k]}, \beta^{[s],[k]}) \in \mathcal{W}$ and $(\alpha^{[t],[k]}, \beta^{[t],[k]}) \in \mathcal{W}$

Compute $N^{[k+1]} := BC_{(\alpha^{[k]},\, \beta^{[k]})}(\mathcal{C}(N^{[k]}))$

Example:

$$\lim_{k\to\infty} N^{[k]}$$

Theorem II (Analysis of convergence)

Let $N^{[0]}$ be a net of C^0 compatible u-functions that are LipC on grid intervals with a bound $\frac{L}{10}$ on the Lipschitz constants. The corner cutting algorithm is convergent for all $(\boldsymbol{\alpha}^{[s],[k]}, \boldsymbol{\beta}^{[s],[k]})$, $(\boldsymbol{\alpha}^{[t],[k]}, \boldsymbol{\beta}^{[t],[k]}) \in \mathcal{W}$ such that

$$\mu = \sup_{k \geq 0} \mu^{[k]} < 1$$

with

$$\mu^{[k]} = \sup_{i \in \mathbb{Z}} \left\{ \begin{array}{l} \beta_i^{[t],[k]} - \alpha_i^{[t],[k]},\ 1 - \beta_{i-1}^{[t],[k]} + \alpha_i^{[t],[k]}, \\ \beta_i^{[s],[k]} - \alpha_i^{[s],[k]},\ 1 - \beta_{i-1}^{[s],[k]} + \alpha_i^{[s],[k]} \end{array} \right\}.$$

Theorem II (Analysis of convergence)

Let $N^{[0]}$ be a net of C^0 compatible u-functions that are LipC on grid intervals with a bound $\frac{L}{10}$ on the Lipschitz constants. The corner cutting algorithm is convergent for all $(\boldsymbol{\alpha}^{[s],[k]}, \boldsymbol{\beta}^{[s],[k]})$, $(\boldsymbol{\alpha}^{[t],[k]}, \boldsymbol{\beta}^{[t],[k]}) \in \mathcal{W}$ such that

$$\mu = \sup_{k \geq 0} \mu^{[k]} < 1$$

with

$$\mu^{[k]} = \sup_{i \in \mathbb{Z}} \left\{ \begin{array}{l} \beta_i^{[t],[k]} - \alpha_i^{[t],[k]},\ 1 - \beta_{i-1}^{[t],[k]} + \alpha_i^{[t],[k]}, \\ \beta_i^{[s],[k]} - \alpha_i^{[s],[k]},\ 1 - \beta_{i-1}^{[s],[k]} + \alpha_i^{[s],[k]} \end{array} \right\}.$$

The proof consists in showing that $\{\mathcal{C}(N^{[k]})\}_{k \geq 0}$ is a Cauchy sequence.

Key steps of the proof:

1. For all $k \geq 0$, $N^{[k+1]} := BC_{(\boldsymbol{\alpha}^{[k]},\, \boldsymbol{\beta}^{[k]})}(\mathcal{C}(N^{[k]}))$ and $\mathcal{C}(N^{[k+1]})$ are both LipC in $\mathbb{R}^2$ with Lipschitz constant L.

Key steps of the proof:

1. For all $k \geq 0$, $N^{[k+1]} := BC_{(\boldsymbol{\alpha}^{[k]},\, \boldsymbol{\beta}^{[k]})}(\mathcal{C}(N^{[k]}))$ and $\mathcal{C}(N^{[k+1]})$ are both LipC in $\mathbb{R}^2$ with Lipschitz constant L.
2. Attaching the u-functions of $N^{[0]}$ to the gridlines of
$$T := T((\mathbf{h}^{[s],[0]}, \mathbf{h}^{[t],[0]}), O) \qquad \text{with} \qquad h_i^{[s],[0]} = h_i^{[t],[0]} = 1,$$
the u-functions of $N^{[k]}$ are attached to the gridlines of $T^{[k]}$ obtained from T by k steps of $CC_{(\boldsymbol{\alpha},\, \boldsymbol{\beta})}$. Therefore:
$$d^{[k+1]} \leq \mu^{[k]}\, d^{[k]} \qquad \text{with} \quad d^{[k]} := \sup_{i \in \mathbb{Z}} \{h_i^{[s],[k]}, h_i^{[t],[k]}\}.$$

Key steps of the proof:

1. For all $k \geq 0$, $N^{[k+1]} := BC_{(\boldsymbol{\alpha}^{[k]},\, \boldsymbol{\beta}^{[k]})}(\mathcal{C}(N^{[k]}))$ and $\mathcal{C}(N^{[k+1]})$ are both LipC in $\mathbb{R}^2$ with Lipschitz constant L.
2. Attaching the u-functions of $N^{[0]}$ to the gridlines of

$$T := T((\mathbf{h}^{[s],[0]}, \mathbf{h}^{[t],[0]}), O) \qquad \text{with} \qquad h_i^{[s],[0]} = h_i^{[t],[0]} = 1,$$

the u-functions of $N^{[k]}$ are attached to the gridlines of $T^{[k]}$ obtained from T by k steps of $CC_{(\boldsymbol{\alpha},\, \boldsymbol{\beta})}$. Therefore:

$$d^{[k+1]} \leq \mu^{[k]}\, d^{[k]} \qquad \text{with} \quad d^{[k]} := \sup_{i \in \mathbb{Z}} \{h_i^{[s],[k]}, h_i^{[t],[k]}\}.$$

Proposition (B)

Let F be a continuous function defined on $R = [a,b] \times [c,d]$ and let $\mathcal{C}(F_{|\partial R})$ be the Coons patch interpolating the u-functions $F(s,c)$, $F(s,d)$, $F(a,t)$, $F(b,t)$. If F is LipC with Lipschitz constant L, then $\|\mathcal{C}(F_{|\partial R}) - F\| \leq 2L \min\{d-c,\, b-a\}$.

Key steps of the proof:

1. For all $k \geq 0$, $N^{[k+1]} := BC_{(\boldsymbol{\alpha}^{[k]},\, \boldsymbol{\beta}^{[k]})}(\mathcal{C}(N^{[k]}))$ and $\mathcal{C}(N^{[k+1]})$ are both LipC in $\mathbb{R}^2$ with Lipschitz constant L.
2. Attaching the u-functions of $N^{[0]}$ to the gridlines of

$$T := T((\mathbf{h}^{[s],[0]}, \mathbf{h}^{[t],[0]}), O) \qquad \text{with} \qquad h_i^{[s],[0]} = h_i^{[t],[0]} = 1,$$

the u-functions of $N^{[k]}$ are attached to the gridlines of $T^{[k]}$ obtained from T by k steps of $CC_{(\boldsymbol{\alpha},\, \boldsymbol{\beta})}$. Therefore:

$$d^{[k+1]} \leq \mu^{[k]}\, d^{[k]} \qquad \text{with} \quad d^{[k]} := \sup_{i \in \mathbb{Z}} \{h_i^{[s],[k]}, h_i^{[t],[k]}\}.$$

Proposition (B)

Let F be a continuous function defined on $R = [a, b] \times [c, d]$ and let $\mathcal{C}(F_{|\partial R})$ be the Coons patch interpolating the u-functions $F(s, c)$, $F(s, d)$, $F(a, t)$, $F(b, t)$. If F is LipC with Lipschitz constant L, then $\|\mathcal{C}(F_{|\partial R}) - F\| \leq 2L \min\{d - c,\ b - a\}$.

$\Downarrow$ regarding $\mathcal{C}(N^{[k+1]})$ as a piecewise Coons approximation of $\mathcal{C}(N^{[k]})$

$$\|\mathcal{C}(N^{[k+1]}) - \mathcal{C}(N^{[k]})\| \leq 2L\, d^{[k]}$$ □

The Generalized Lane-Riesenfeld algorithm for nets of functions

The GLR-algorithm for nets

Input: compatible net $N^{[0]}$

For $k = 0, 1, \dots$

Input: $\alpha^{[k]}$ and $\beta^{[k]}$

Compute $N^{[k+1,0]} := BC_{(\alpha^{[k]},\,\beta^{[k]})}(\mathcal{C}(N^{[k]}))$

Input: $w_i^{[s],[k]}, w_j^{[t],[k]} \in (0,1)\ \forall i,j \in \mathbb{Z},\ k \geq 0$

For $j = 0, ..., m_k - 1 \quad (m_k \in \mathbb{N}_0\ u.b.)$

$$N^{[k+1,j+1]} = A^{[k]}(\mathcal{C}(N^{[k+1,j]}))$$

$$\text{with } A(\mathcal{C}(N)) = \{\mathcal{C}(N)(s, t_j + w_j^{[t]} h_j^{[t]}),\ j \in \mathbb{Z}\} \cup \{\mathcal{C}(N)(s_i + w_i^{[s]} h_i^{[s]}, t),\ i \in \mathbb{Z}\}$$

$N^{[k+1]} = N^{[k+1,m_k]}$

☞ The GLR-algorithm converges!

The Generalized Lane-Riesenfeld algorithm for nets of functions

Example: $m_k = 1 \;\; \forall k \geq 0$

The GLR-algorithm for nets

Input: compatible net $N^{[0]}$

For $\;k = 0, 1, \ldots$

Input: $\alpha^{[k]}$ and $\beta^{[k]}$

Compute $N^{[k+1,0]} := BC_{(\alpha^{[k]},\, \beta^{[k]})}(\mathcal{C}(N^{[k]}))$

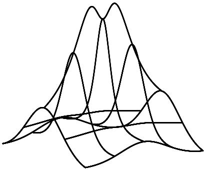
Input: $w_i^{[s],[k]}, w_j^{[t],[k]} \in (0,1) \; \forall i,j \in \mathbb{Z},\; k \geq 0$

For $j = 0, ..., m_k - 1 \;\; (m_k \in \mathbb{N}_0 \; u.b.)$

$$N^{[k+1,j+1]} = A^{[k]}(\mathcal{C}(N^{[k+1,j]}))$$

with $A(\mathcal{C}(N)) = \{\mathcal{C}(N)(s, t_j + w_j^{[t]} h_j^{[t]}),\; j \in \mathbb{Z}\} \cup \{\mathcal{C}(N)(s_i + w_i^{[s]} h_i^{[s]}, t),\; i \in \mathbb{Z}\}$

$N^{[k+1]} = N^{[k+1, m_k]}$


$N^{[0]}$

☞ The GLR-algorithm converges!

The Generalized Lane-Riesenfeld algorithm for nets of functions

The GLR-algorithm for nets

Input: compatible net $N^{[0]}$

For $k = 0, 1, \ldots$

Input: $\alpha^{[k]}$ and $\beta^{[k]}$

Compute $N^{[k+1,0]} := BC_{(\alpha^{[k]},\,\beta^{[k]})}(\mathcal{C}(N^{[k]}))$

Input: $w_i^{[s],[k]}, w_j^{[t],[k]} \in (0,1)\ \forall i,j \in \mathbb{Z},\ k \geq 0$

For $j = 0, \ldots, m_k - 1 \quad (m_k \in \mathbb{N}_0\ u.b.)$

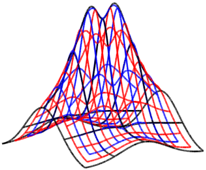
$$N^{[k+1,j+1]} = A^{[k]}(\mathcal{C}(N^{[k+1,j]}))$$

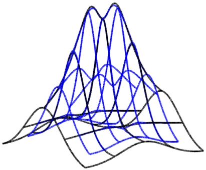
with $A(\mathcal{C}(N)) = \{\mathcal{C}(N)(s, t_j + w_j^{[t]} h_j^{[t]}),\ j \in \mathbb{Z}\}$
$\cup \{\mathcal{C}(N)(s_i + w_i^{[s]} h_i^{[s]}, t),\ i \in \mathbb{Z}\}$

$N^{[k+1]} = N^{[k+1,m_k]}$

☞ The GLR-algorithm converges!

Example: $m_k = 1 \quad \forall k \geq 0$





$N^{[1]}$

The Generalized Lane-Riesenfeld algorithm for nets of functions

Example: $m_k = 1 \ \ \forall k \geq 0$

The GLR-algorithm for nets

Input: compatible net $N^{[0]}$

For $k = 0, 1, \dots$

Input: $\boldsymbol{\alpha}^{[k]}$ and $\boldsymbol{\beta}^{[k]}$

Compute $N^{[k+1,0]} := BC_{(\boldsymbol{\alpha}^{[k]},\, \boldsymbol{\beta}^{[k]})}(\mathcal{C}(N^{[k]}))$

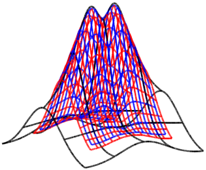
Input: $w_i^{[s],[k]}, w_j^{[t],[k]} \in (0,1)\ \forall i,j \in \mathbb{Z},\ k \geq 0$

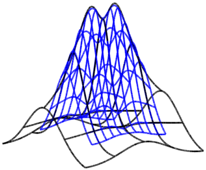
For $j = 0, ..., m_k - 1 \ \ (m_k \in \mathbb{N}_0\ u.b.)$

$$N^{[k+1,j+1]} = A^{[k]}(\mathcal{C}(N^{[k+1,j]}))$$

with $A(\mathcal{C}(N)) = \{\mathcal{C}(N)(s, t_j + w_j^{[t]} h_j^{[t]}),\ j \in \mathbb{Z}\}$
$\cup \{\mathcal{C}(N)(s_i + w_i^{[s]} h_i^{[s]}, t),\ i \in \mathbb{Z}\}$

$N^{[k+1]} = N^{[k+1,m_k]}$





$N^{[2]}$

☞ The GLR-algorithm converges!

The Generalized Lane-Riesenfeld algorithm for nets of functions

Example: $m_k = 1 \ \ \forall k \geq 0$

The GLR-algorithm for nets

Input: compatible net $N^{[0]}$

For $\ k = 0, 1, \ldots$

Input: $\boldsymbol{\alpha}^{[k]}$ and $\boldsymbol{\beta}^{[k]}$

Compute $N^{[k+1,0]} := BC_{(\boldsymbol{\alpha}^{[k]},\, \boldsymbol{\beta}^{[k]})}(\mathcal{C}(N^{[k]}))$

Input: $w_i^{[s],[k]}, w_j^{[t],[k]} \in (0,1) \ \forall i, j \in \mathbb{Z}, \ k \geq 0$

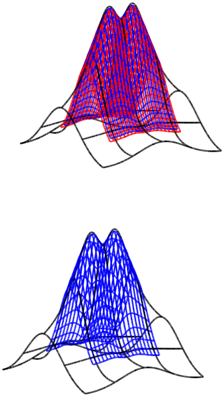
For $j = 0, \ldots, m_k - 1 \ \ (m_k \in \mathbb{N}_0 \ u.b.)$

$$N^{[k+1,j+1]} = A^{[k]}(\mathcal{C}(N^{[k+1,j]}))$$

with $A(\mathcal{C}(N)) = \{\mathcal{C}(N)(s, t_j + w_j^{[t]} h_j^{[t]}),\ j \in \mathbb{Z}\} \cup \{\mathcal{C}(N)(s_i + w_i^{[s]} h_i^{[s]}, t),\ i \in \mathbb{Z}\}$

$$N^{[k+1]} = N^{[k+1,m_k]}$$

☞ The GLR-algorithm converges!



$N^{[3]}$

The Generalized Lane-Riesenfeld algorithm for nets of functions

The GLR-algorithm for nets

Input: compatible net $N^{[0]}$

For $k = 0, 1, \ldots$

Input: $\boldsymbol{\alpha}^{[k]}$ and $\boldsymbol{\beta}^{[k]}$

Compute $N^{[k+1,0]} := BC_{(\boldsymbol{\alpha}^{[k]},\, \boldsymbol{\beta}^{[k]})}(\mathcal{C}(N^{[k]}))$

Input: $w_i^{[s],[k]}, w_j^{[t],[k]} \in (0,1)\ \forall i, j \in \mathbb{Z},\ k \geq 0$

For $j = 0, \ldots, m_k - 1 \ \ (m_k \in \mathbb{N}_0\ u.b.)$

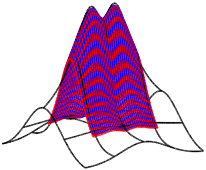
$$N^{[k+1,j+1]} = A^{[k]}(\mathcal{C}(N^{[k+1,j]}))$$

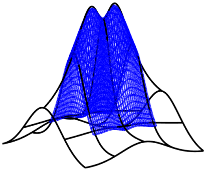
with $A(\mathcal{C}(N)) = \{\mathcal{C}(N)(s, t_j + w_j^{[t]} h_j^{[t]}),\ j \in \mathbb{Z}\} \cup \{\mathcal{C}(N)(s_i + w_i^{[s]} h_i^{[s]}, t),\ i \in \mathbb{Z}\}$

$N^{[k+1]} = N^{[k+1,m_k]}$

☞ The GLR-algorithm converges!

Example: $m_k = 1 \ \ \forall k \geq 0$





$N^{[4]}$

The Generalized Lane-Riesenfeld algorithm for nets of functions

The GLR-algorithm for nets

Input: compatible net $N^{[0]}$

For $k = 0, 1, \ldots$

Input: $\boldsymbol{\alpha}^{[k]}$ and $\boldsymbol{\beta}^{[k]}$

Compute $N^{[k+1,0]} := BC_{(\boldsymbol{\alpha}^{[k]},\, \boldsymbol{\beta}^{[k]})}(\mathcal{C}(N^{[k]}))$

Input: $w_i^{[s],[k]}, w_j^{[t],[k]} \in (0,1)\ \forall i, j \in \mathbb{Z},\ k \geq 0$

For $j = 0, ..., m_k - 1 \quad (m_k \in \mathbb{N}_0\ u.b.)$

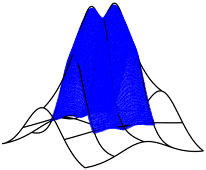
$$N^{[k+1,j+1]} = A^{[k]}(\mathcal{C}(N^{[k+1,j]}))$$

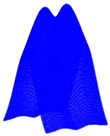
with $A(\mathcal{C}(N)) = \{\mathcal{C}(N)(s, t_j + w_j^{[t]} h_j^{[t]}),\ j \in \mathbb{Z}\} \cup \{\mathcal{C}(N)(s_i + w_i^{[s]} h_i^{[s]}, t),\ i \in \mathbb{Z}\}$

$N^{[k+1]} = N^{[k+1,m_k]}$

☞ The GLR-algorithm converges!

Example: $m_k = 1 \quad \forall k \geq 0$





$\lim_{k\to\infty} N^{[k]}$


danke
謝謝
ngiyabonga
teşekkür ederim
спасибо
thank you
gracias
tapadh leat
bedankt
dziękuję
mochchakkeram
go raibh maith agat
obrigado
sukriya
kop khun krap
arigato
dakujem
мерси
terima kasih
grazie
merci
감사합니다
dank je
hvala

SMART 2017

17th – 21st September, 2017 Gaeta, Italy

2nd International Conference on Subdivision; Geometric and Algebraic Methods, Isogeometric Analysis and Refinability in ITaly

Web site: sbai.uniroma1.it/smart2017

Topics

Topics include Algebraic and Differential Geometry, Computer Aided Design, Curve and Surface Design, Finite Elements, NURBS and Isogeometric Analysis, Refinability, Approximation Theory, Subdivision, Wavelets and Multiresolution Methods….

Organizing Committee

Costanza Conti (Univ. Firenze)
Mariantonia Cotronei (Univ. Reggio Calabria)
Serena Morigi (Univ. Bologna)
Enza Pellegrino (Univ. L'Aquila)
Francesca Pelosi (Univ. Roma "Tor Vergata")
Francesca Pitolli (Univ. Roma "La Sapienza")
Sara Remogna (Univ. Torino)
Lucia Romani (Univ. Milano-Bicocca)
Maria Lucia Sampoli (Univ. Siena)
Alessandra Sestini (Univ. Firenze)

Venue

Venue: Hotel Serapo
Located on the slopes of the Natural Park of Monte Orlando in the most beautiful and panoramic corner overlooking Serapo Beach, very close to the city center and the old town of Gaeta.