King Abdullah University of Science and Technology

KAUST

# CS 247 – Scientific Visualization
# Lecture 12: Scalar Fields, Pt. 8

Markus Hadwiger, KAUST

# Reading Assignment #6 (until Mar 8)

Read (required):

- Real-Time Volume Graphics, Chapter 2
  (*GPU Programming*)
- Real-Time Volume Graphics, Chapters 5.5 and 5.6 (you already had to read - 5.4)
  (*Local Volume Illumination*)
- Refresh your memory on eigenvectors and eigenvalues:
  **`https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors`**

# Metric Tensor (Field)

Symmetric, covariant second-order tensor field:
*defines* inner product on manifold (in each tangent space)

$$\langle \mathbf{v}, \mathbf{w} \rangle := \mathbf{g}(\mathbf{v}, \mathbf{w}) = \|\mathbf{v}\| \, \|\mathbf{w}\| \cos\theta$$

# Metric Tensor (Field)

Symmetric, covariant second-order tensor field:

*defines* inner product on manifold (in each tangent space)

$$\langle \mathbf{v}, \mathbf{w} \rangle := \mathbf{g}(\mathbf{v}, \mathbf{w}) = \|\mathbf{v}\| \, \|\mathbf{w}\| \cos\theta$$

$$\begin{aligned} \|\mathbf{v}\|^2 &= \langle \mathbf{v}, \mathbf{v} \rangle \\ &= \mathbf{g}(\mathbf{v}, \mathbf{v}) \\ &= g_{ij} v^i v^j \\ &= \mathbf{v}^T \mathbf{g} \, \mathbf{v} \end{aligned}$$

# Metric Tensor (Field)

Symmetric, covariant second-order tensor field:

*defines* inner product on manifold (in each tangent space)

$$\langle \mathbf{v}, \mathbf{w} \rangle := \mathbf{g}(\mathbf{v}, \mathbf{w}) = \|\mathbf{v}\| \, \|\mathbf{w}\| \cos\theta$$

$$g_{ij} = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}$$

(2D)

$$\begin{aligned} \|\mathbf{v}\|^2 &= \langle \mathbf{v}, \mathbf{v} \rangle \\ &= \mathbf{g}(\mathbf{v}, \mathbf{v}) \\ &= g_{ij} v^i v^j \\ &= \mathbf{v}^T \mathbf{g} \, \mathbf{v} \end{aligned}$$

$$\|\mathbf{v}\|^2 = \begin{bmatrix} v^1 & v^2 \end{bmatrix} \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} v^1 \\ v^2 \end{bmatrix}$$

# Metric Tensor (Field)

Symmetric, covariant second-order tensor field:
*defines* inner product on manifold (in each tangent space)

$$\langle \mathbf{v}, \mathbf{w} \rangle := \mathbf{g}(\mathbf{v}, \mathbf{w}) = \|\mathbf{v}\| \|\mathbf{w}\| \cos\theta$$

$$g_{ij} = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}$$

(2D)

$$\begin{aligned} \|\mathbf{v}\|^2 &= \langle \mathbf{v}, \mathbf{v} \rangle \\ &= \mathbf{g}(\mathbf{v}, \mathbf{v}) \\ &= g_{ij} v^i v^j \\ &= \mathbf{v}^T \mathbf{g} \, \mathbf{v} \end{aligned}$$

$$\|\mathbf{v}\|^2 = \begin{bmatrix} v^1 & v^2 \end{bmatrix} \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} v^1 \\ v^2 \end{bmatrix}$$

Cartesian coordinates: $g_{ij} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

$$\|\mathbf{v}\|^2 = \begin{bmatrix} v^1 & v^2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v^1 \\ v^2 \end{bmatrix} = \mathbf{v}^T \mathbf{v}$$

# Metric Tensor (Field)

Components of metric referred to coordinates

$$g_{ij} := \langle \mathbf{e}_i, \mathbf{e}_j \rangle$$

A second-order tensor field is bi-linear, i.e.,
linear in each (vector/covector) argument separately

From bi-linearity we immediately get:

$$\begin{aligned}
\langle \mathbf{v}, \mathbf{w} \rangle &= \mathbf{g}\left(v^i \mathbf{e}_i, w^j \mathbf{e}_j\right) \\
&= v^i w^j \langle \mathbf{e}_i, \mathbf{e}_j \rangle \\
&= g_{ij} v^i w^j
\end{aligned}$$

# Gradient Vector from Differential 1-Form

The metric (and inverse metric) *lower* or *raise* indices
(i.e., convert between covariant and contravariant tensors)

$$v^i = g^{ij} v_j$$
$$v_i = g_{ij} v^j$$

$$v^i \mathbf{e}_i = g^{ij} v_j \mathbf{e}_i$$
$$v_i \boldsymbol{\omega}^i = g_{ij} v^j \boldsymbol{\omega}^i$$

Inverse metric (contravariant)

$$[g^{ij}] = [g_{ij}]^{-1}$$

$$g^{ik} g_{kj} = \delta^i_j$$

Kronecker delta behaves
like identity matrix

# Gradient Vector from Differential 1-Form

So the gradient vector is

$$\nabla f = \left( g^{ij} \frac{\partial f}{\partial x^j} \right) \mathbf{e}_i$$

*Vector-valued* 1-form

$$d\mathbf{r} = dx^i \, \mathbf{e}_i$$

$$d\mathbf{r}(\cdot) = dx^i(\cdot) \, \mathbf{e}_i$$

Directional derivative via inner product:

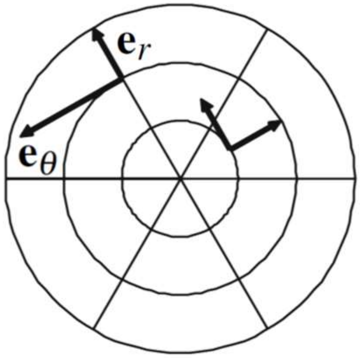
$$\begin{aligned} \langle \nabla f, \cdot \rangle &= g_{kj} g^{ik} \frac{\partial f}{\partial x^i} dx^j(\cdot) \\ &= \delta^i{}_j \frac{\partial f}{\partial x^i} dx^j(\cdot) \\ &= \frac{\partial f}{\partial x^i} dx^i(\cdot) \end{aligned}$$

$$\begin{aligned} \nabla f \cdot d\mathbf{r} &= g_{kj} g^{ik} \frac{\partial f}{\partial x^i} dx^j \\ &= \delta^i{}_j \frac{\partial f}{\partial x^i} dx^j \\ &= \frac{\partial f}{\partial x^i} dx^i \end{aligned}$$

# Example: Polar Coordinates

Metric tensor and inverse metric for polar coordinates

$$g_{ij} = \begin{bmatrix} 1 & 0 \\ 0 & r^2 \end{bmatrix} \qquad g^{ij} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{r^2} \end{bmatrix}$$



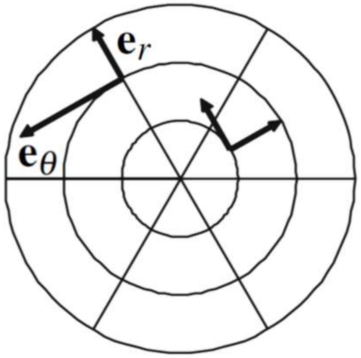

Gradient vector from 1-form: raise index with inverse metric

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial r} & \frac{\partial f}{\partial \theta} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{r^2} \end{bmatrix} \begin{bmatrix} \mathbf{e}_r \\ \mathbf{e}_\theta \end{bmatrix} = \frac{\partial f}{\partial r}\mathbf{e}_r + \frac{1}{r^2}\frac{\partial f}{\partial \theta}\mathbf{e}_\theta$$

# Example: Polar Coordinates

Metric tensor and inverse metric for polar coordinates

$$g_{ij} = \begin{bmatrix} 1 & 0 \\ 0 & r^2 \end{bmatrix} \qquad g^{ij} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{r^2} \end{bmatrix}$$




Gradient vector from 1-form: raise index with inverse metric

$$\nabla f(r,\theta) = \frac{\partial f(r,\theta)}{\partial r}\,\mathbf{e}_r(r,\theta) + \frac{1}{r^2}\frac{\partial f(r,\theta)}{\partial \theta}\,\mathbf{e}_\theta(r,\theta)$$

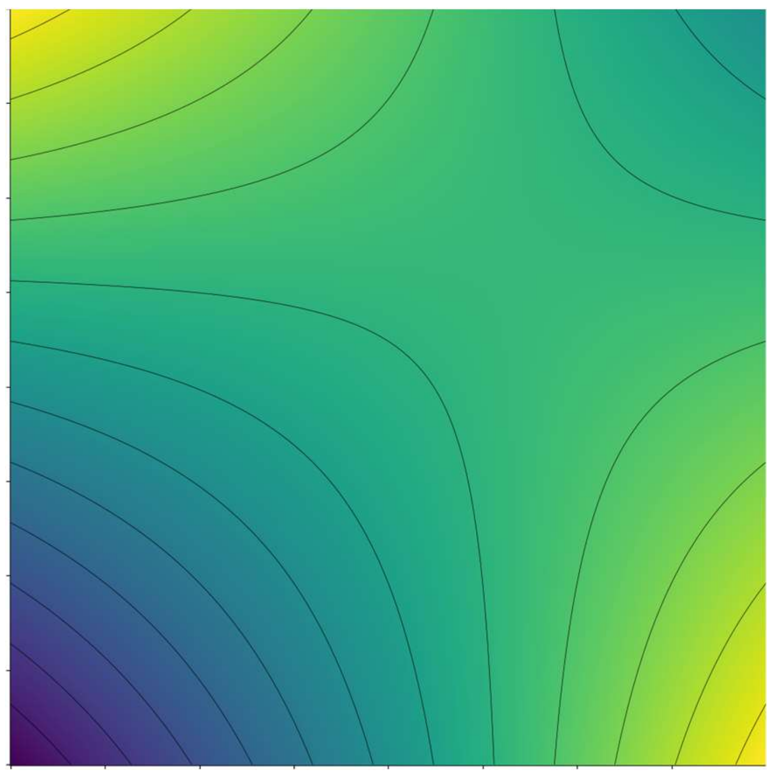
don’t forget that all of this is position-dependent!

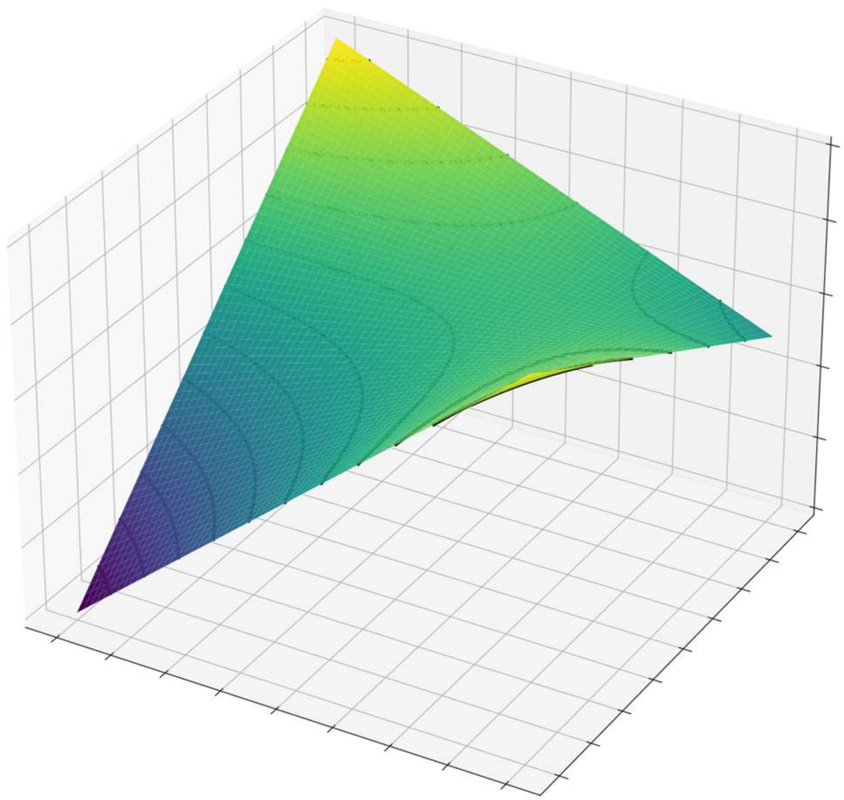
# Multi-Linear Interpolation

# Bi-Linear Interpolation

Consider area between 2x2 adjacent samples (e.g., pixel centers)

Example #2: 1 at top-left and bottom-right, 0 at bottom-left, 0.5 at top-right





# Bi-Linear Interpolation

Consider area between 2x2 adjacent samples (e.g., pixel centers):

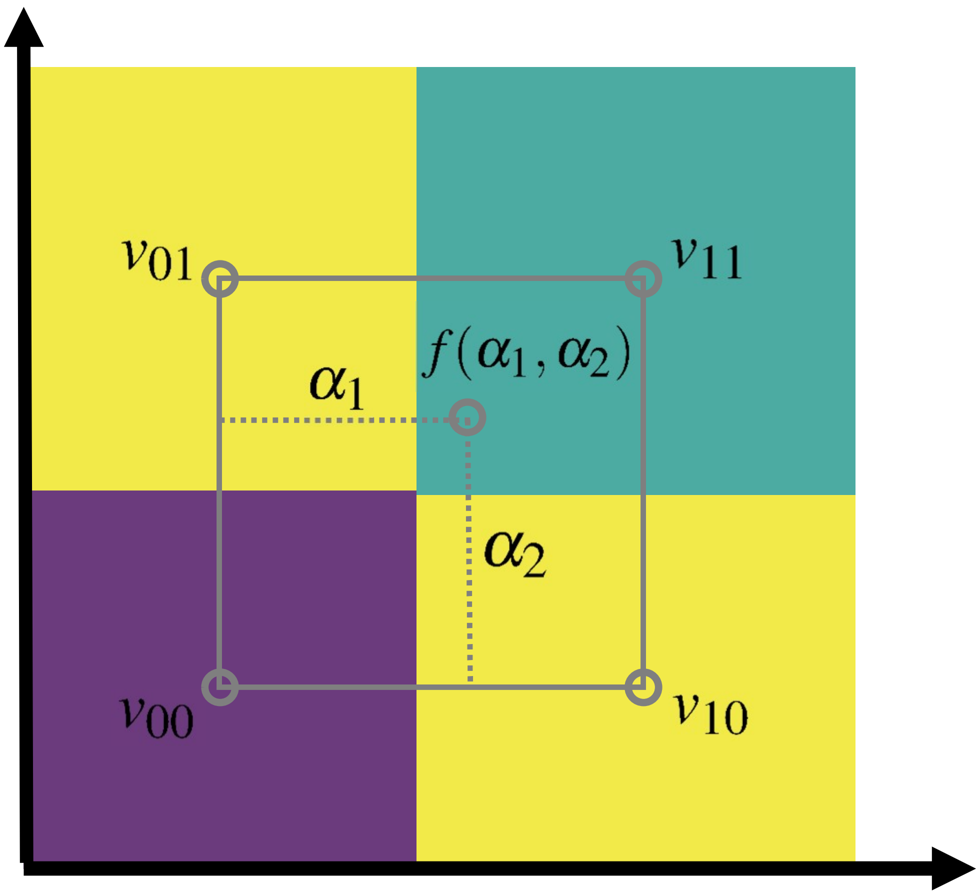
Given any (fractional) position

$$\alpha_1 := x_1 - \lfloor x_1 \rfloor \qquad \alpha_1 \in [0.0, 1.0)$$
$$\alpha_2 := x_2 - \lfloor x_2 \rfloor \qquad \alpha_2 \in [0.0, 1.0)$$

and 2x2 sample values

$$\begin{bmatrix} v_{01} & v_{11} \\ v_{00} & v_{10} \end{bmatrix}$$

Compute: $f(\alpha_1, \alpha_2)$



# Bi-Linear Interpolation

Consider area between 2x2 adjacent samples (e.g., pixel centers):

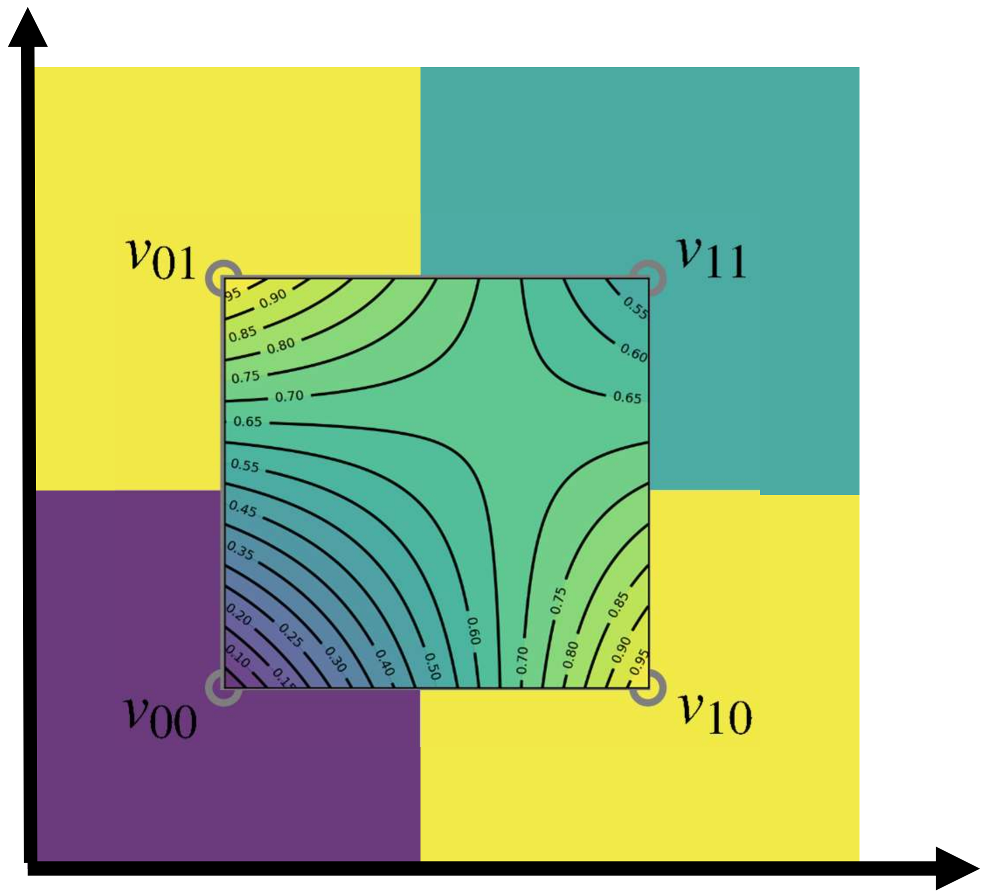
Given any (fractional) position

$$\alpha_1 := x_1 - \lfloor x_1 \rfloor \qquad \alpha_1 \in [0.0, 1.0)$$
$$\alpha_2 := x_2 - \lfloor x_2 \rfloor \qquad \alpha_2 \in [0.0, 1.0)$$

and 2x2 sample values

$$\begin{bmatrix} v_{01} & v_{11} \\ v_{00} & v_{10} \end{bmatrix}$$

Compute: $f(\alpha_1, \alpha_2)$



# Bi-Linear Interpolation

Interpolate function at (fractional) position $(\alpha_1, \alpha_2)$ :

$$f(\alpha_1, \alpha_2) = \begin{bmatrix} \alpha_2 & (1-\alpha_2) \end{bmatrix} \begin{bmatrix} v_{01} & v_{11} \\ v_{00} & v_{10} \end{bmatrix} \begin{bmatrix} (1-\alpha_1) \\ \alpha_1 \end{bmatrix}$$

$$= \begin{bmatrix} \alpha_2 & (1-\alpha_2) \end{bmatrix} \begin{bmatrix} (1-\alpha_1)v_{01} + \alpha_1 v_{11} \\ (1-\alpha_1)v_{00} + \alpha_1 v_{10} \end{bmatrix}$$

$$= \begin{bmatrix} \alpha_2 v_{01} + (1-\alpha_2)v_{00} & \alpha_2 v_{11} + (1-\alpha_2)v_{10} \end{bmatrix} \begin{bmatrix} (1-\alpha_1) \\ \alpha_1 \end{bmatrix}$$

# Bi-Linear Interpolation

Interpolate function at (fractional) position $(\alpha_1, \alpha_2)$ :

$$f(\alpha_1, \alpha_2) = \begin{bmatrix} \alpha_2 & (1-\alpha_2) \end{bmatrix} \begin{bmatrix} v_{01} & v_{11} \\ v_{00} & v_{10} \end{bmatrix} \begin{bmatrix} (1-\alpha_1) \\ \alpha_1 \end{bmatrix}$$

$$= (1-\alpha_1)(1-\alpha_2)v_{00} + \alpha_1(1-\alpha_2)v_{10} + (1-\alpha_1)\alpha_2 v_{01} + \alpha_1 \alpha_2 v_{11}$$

$$= v_{00} + \alpha_1(v_{10} - v_{00}) + \alpha_2(v_{01} - v_{00}) + \alpha_1 \alpha_2(v_{00} + v_{11} - v_{10} - v_{01})$$

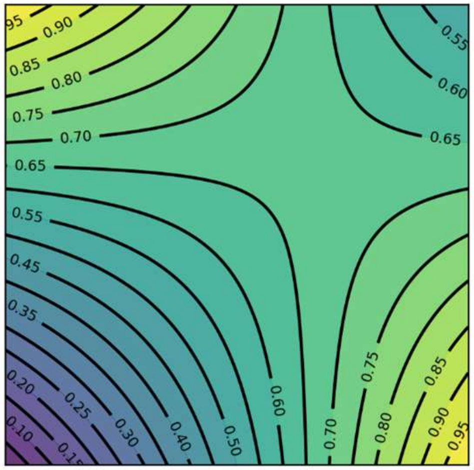
# Bi-Linear Interpolation: Contours

Find one specific iso-contour (can of course do this for any/all isovalues):

$$f(\alpha_1, \alpha_2) = c$$

Find all $(\alpha_1, \alpha_2)$ where:

$$v_{00} + \alpha_1 (v_{10} - v_{00}) + \alpha_2 (v_{01} - v_{00}) + \alpha_1 \alpha_2 (v_{00} + v_{11} - v_{10} - v_{01}) = c$$



# Bi-Linear Interpolation: Critical Points

Compute gradient (critical points are where gradient is zero vector):

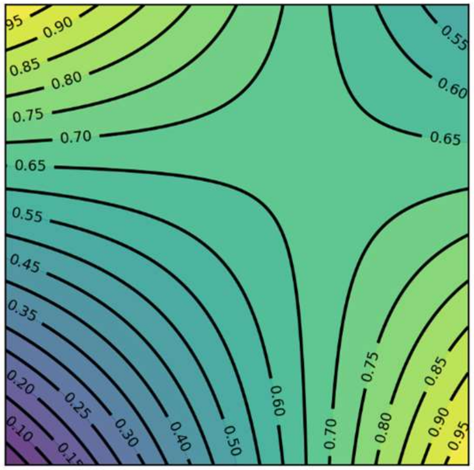
$$\frac{\partial f(\alpha_1,\alpha_2)}{\partial \alpha_1} = (v_{10} - v_{00}) + \alpha_2 (v_{00} + v_{11} - v_{10} - v_{01})$$

$$\frac{\partial f(\alpha_1,\alpha_2)}{\partial \alpha_2} = (v_{01} - v_{00}) + \alpha_1 (v_{00} + v_{11} - v_{10} - v_{01})$$

Where are lines of constant value / critical points?

$$\frac{\partial f(\alpha_1,\alpha_2)}{\partial \alpha_1} = 0: \qquad \alpha_2 = \frac{v_{00} - v_{10}}{v_{00} + v_{11} - v_{10} - v_{01}}$$

$$\frac{\partial f(\alpha_1,\alpha_2)}{\partial \alpha_2} = 0: \qquad \alpha_1 = \frac{v_{00} - v_{01}}{v_{00} + v_{11} - v_{10} - v_{01}}$$



# Bi-Linear Interpolation: Critical Points

Compute gradient (critical points are where gradient is zero vector):

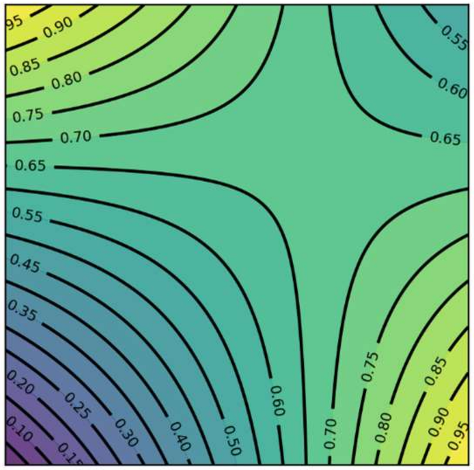
$$\frac{\partial f(\alpha_1, \alpha_2)}{\partial \alpha_1} = (v_{10} - v_{00}) + \alpha_2 (v_{00} + v_{11} - v_{10} - v_{01})$$

$$\frac{\partial f(\alpha_1, \alpha_2)}{\partial \alpha_2} = (v_{01} - v_{00}) + \alpha_1 (v_{00} + v_{11} - v_{10} - v_{01})$$

Where are lines of constant value / critical points?

$$\frac{\partial f(\alpha_1, \alpha_2)}{\partial \alpha_1} = 0: \qquad \alpha_2 = \frac{v_{00} - v_{10}}{v_{00} + v_{11} - v_{10} - v_{01}}$$

$$\frac{\partial f(\alpha_1, \alpha_2)}{\partial \alpha_2} = 0: \qquad \alpha_1 = \frac{v_{00} - v_{01}}{v_{00} + v_{11} - v_{10} - v_{01}}$$




if denominator is zero, bi-linear interpolation has degenerated to linear interpolation (or const)! (also means: no isolated critical points!)

# Bi-Linear Interpolation: Critical Points

Compute gradient (critical points are where gradient is zero vector):

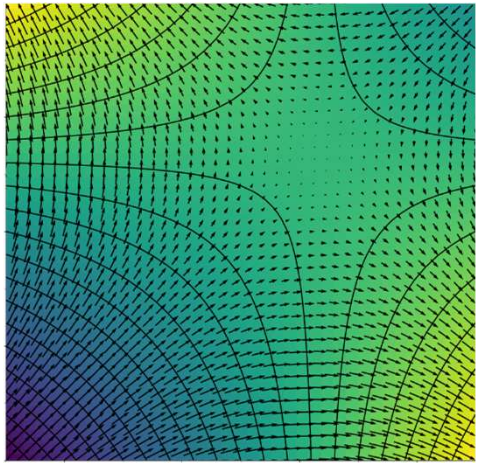
$$\frac{\partial f(\alpha_1,\alpha_2)}{\partial \alpha_1} = (v_{10} - v_{00}) + \alpha_2 (v_{00} + v_{11} - v_{10} - v_{01})$$

$$\frac{\partial f(\alpha_1,\alpha_2)}{\partial \alpha_2} = (v_{01} - v_{00}) + \alpha_1 (v_{00} + v_{11} - v_{10} - v_{01})$$

Where are lines of constant value / critical points?

$$\frac{\partial f(\alpha_1,\alpha_2)}{\partial \alpha_1} = 0: \qquad \alpha_2 = \frac{v_{00} - v_{10}}{v_{00} + v_{11} - v_{10} - v_{01}}$$

$$\frac{\partial f(\alpha_1,\alpha_2)}{\partial \alpha_2} = 0: \qquad \alpha_1 = \frac{v_{00} - v_{01}}{v_{00} + v_{11} - v_{10} - v_{01}}$$



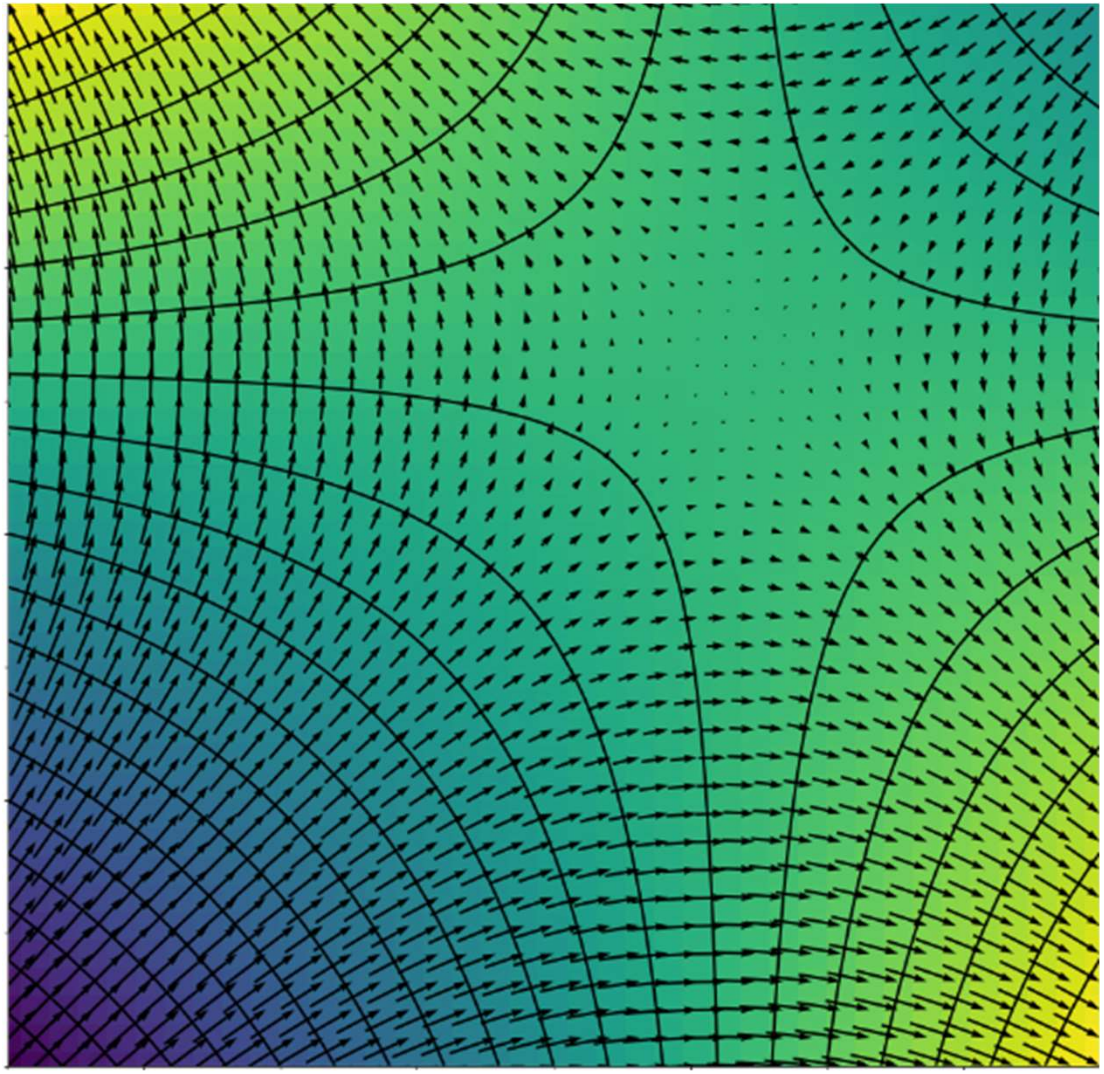
# Bi-Linear Interpolation: Critical Points

Compute gradient

Note that isolines are
farther apart where
gradient is smaller

Note the horizontal and
vertical lines where
gradient becomes
vertical/horizontal

Note the critical point



# Bi-Linear Interpolation: Critical Points

Compute gradient (critical points are where gradient is zero vector):

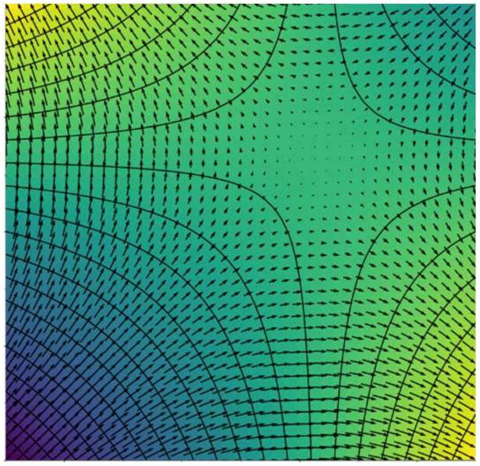
$$\frac{\partial f(\alpha_1,\alpha_2)}{\partial \alpha_1} = (v_{10} - v_{00}) + \alpha_2 (v_{00} + v_{11} - v_{10} - v_{01})$$

$$\frac{\partial f(\alpha_1,\alpha_2)}{\partial \alpha_2} = (v_{01} - v_{00}) + \alpha_1 (v_{00} + v_{11} - v_{10} - v_{01})$$

Where are lines of constant value / critical points?

$$\frac{\partial f(\alpha_1,\alpha_2)}{\partial \alpha_1} = 0: \qquad \alpha_2 = \frac{v_{00} - v_{10}}{v_{00} + v_{11} - v_{10} - v_{01}}$$

$$\frac{\partial f(\alpha_1,\alpha_2)}{\partial \alpha_2} = 0: \qquad \alpha_1 = \frac{v_{00} - v_{01}}{v_{00} + v_{11} - v_{10} - v_{01}}$$



# Bi-Linear Interpolation: Critical Points

Compute gradient (critical points are where gradient is zero vector):

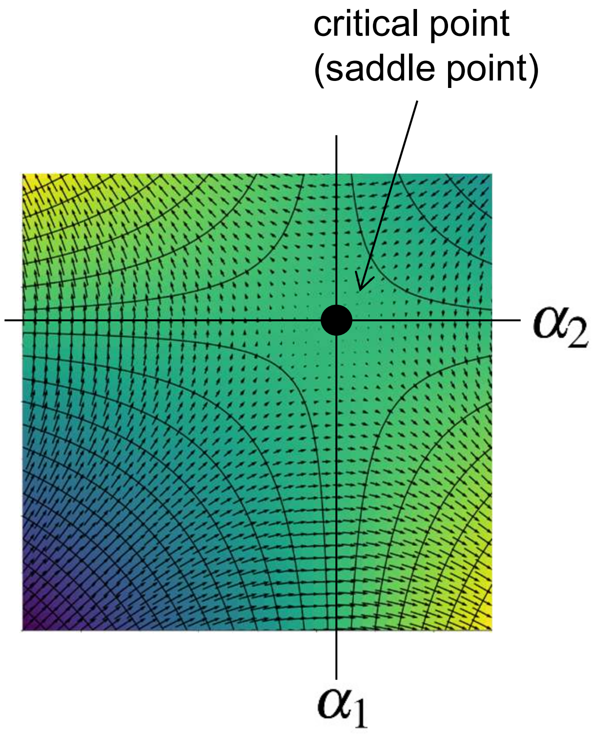
$$\frac{\partial f(\alpha_1,\alpha_2)}{\partial \alpha_1} = (v_{10} - v_{00}) + \alpha_2(v_{00} + v_{11} - v_{10} - v_{01})$$

$$\frac{\partial f(\alpha_1,\alpha_2)}{\partial \alpha_2} = (v_{01} - v_{00}) + \alpha_1(v_{00} + v_{11} - v_{10} - v_{01})$$

Where are lines of constant value / critical points?

$$\frac{\partial f(\alpha_1,\alpha_2)}{\partial \alpha_1} = 0: \quad \alpha_2 = \frac{v_{00} - v_{10}}{v_{00} + v_{11} - v_{10} - v_{01}}$$

$$\frac{\partial f(\alpha_1,\alpha_2)}{\partial \alpha_2} = 0: \quad \alpha_1 = \frac{v_{00} - v_{01}}{v_{00} + v_{11} - v_{10} - v_{01}}$$



# Bi-Linear Interpolation: Critical Points

Compute gradient (critical points are where gradient is zero vector):

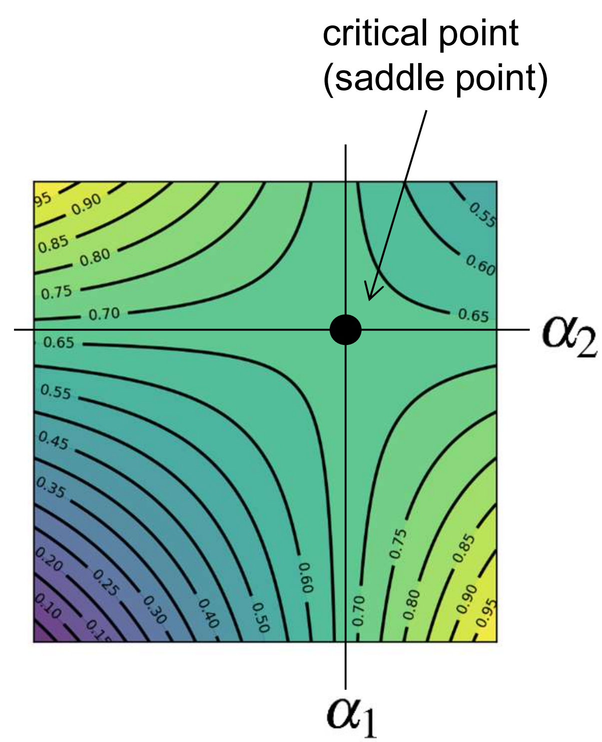
$$\frac{\partial f(\alpha_1,\alpha_2)}{\partial \alpha_1} = (v_{10} - v_{00}) + \alpha_2 (v_{00} + v_{11} - v_{10} - v_{01})$$

$$\frac{\partial f(\alpha_1,\alpha_2)}{\partial \alpha_2} = (v_{01} - v_{00}) + \alpha_1 (v_{00} + v_{11} - v_{10} - v_{01})$$

Where are lines of constant value / critical points?

$$\frac{\partial f(\alpha_1,\alpha_2)}{\partial \alpha_1} = 0 : \qquad \alpha_2 = \frac{v_{00} - v_{10}}{v_{00} + v_{11} - v_{10} - v_{01}}$$

$$\frac{\partial f(\alpha_1,\alpha_2)}{\partial \alpha_2} = 0 : \qquad \alpha_1 = \frac{v_{00} - v_{01}}{v_{00} + v_{11} - v_{10} - v_{01}}$$



# Bi-Linear Interpolation: Critical Points

Examine Hessian matrix at critical point (non-degenerate critical p.?, ...)

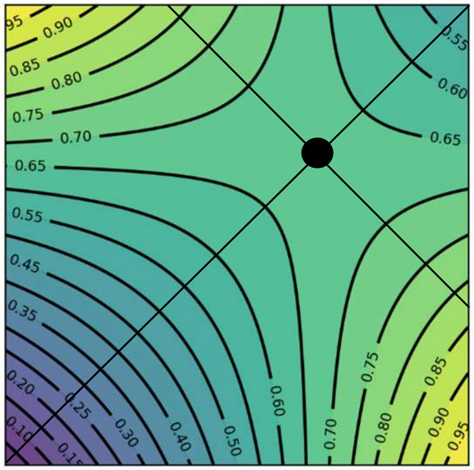
$$\begin{bmatrix} \frac{\partial^2 f}{\partial \alpha_1^2} & \frac{\partial^2 f}{\partial \alpha_1 \partial \alpha_2} \\ \frac{\partial^2 f}{\partial \alpha_2 \partial \alpha_1} & \frac{\partial^2 f}{\partial \alpha_2^2} \end{bmatrix} = \begin{bmatrix} 0 & a \\ a & 0 \end{bmatrix}$$

$$a = v_{00} + v_{11} - v_{10} - v_{01}$$

Eigenvalues and eigenvectors (Hessian is symmetric: always real)

$$\lambda_1 = -a \text{ and } \lambda_2 = a$$

$$v_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, v_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$




(here also: principal curvature magnitudes and directions
of this function's graph == surface embedded in 3D)

# Bi-Linear Interpolation: Critical Points

Examine Hessian matrix at critical point (non-degenerate critical p.?, ...)

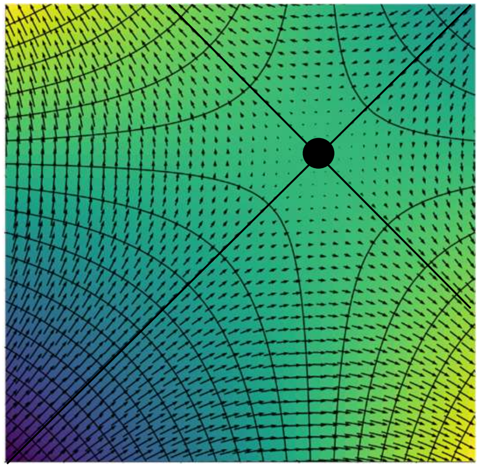
$$\begin{bmatrix} \frac{\partial^2 f}{\partial \alpha_1^2} & \frac{\partial^2 f}{\partial \alpha_1 \partial \alpha_2} \\ \frac{\partial^2 f}{\partial \alpha_2 \partial \alpha_1} & \frac{\partial^2 f}{\partial \alpha_2^2} \end{bmatrix} = \begin{bmatrix} 0 & a \\ a & 0 \end{bmatrix}$$

$$a = v_{00} + v_{11} - v_{10} - v_{01}$$

Eigenvalues and eigenvectors (Hessian is symmetric: always real)

$$\lambda_1 = -a \text{ and } \lambda_2 = a$$

$$v_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, v_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$



(here also: principal curvature magnitudes and directions of this function's graph == surface embedded in 3D)

# Bi-Linear Interpolation: Critical Points

Examine Hessian matrix at critical point (non-degenerate critical p.?, ...)

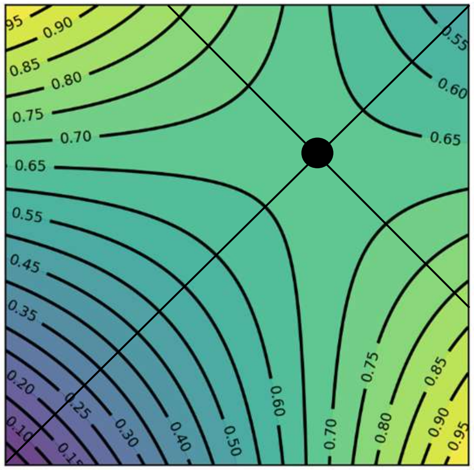
$$\begin{bmatrix} \frac{\partial^2 f}{\partial \alpha_1^2} & \frac{\partial^2 f}{\partial \alpha_1 \partial \alpha_2} \\ \frac{\partial^2 f}{\partial \alpha_2 \partial \alpha_1} & \frac{\partial^2 f}{\partial \alpha_2^2} \end{bmatrix} = \begin{bmatrix} 0 & a \\ a & 0 \end{bmatrix}$$

$$a = v_{00} + v_{11} - v_{10} - v_{01}$$

Eigenvalues and eigenvectors (Hessian is symmetric: always real)

$$\lambda_1 = -a \text{ and } \lambda_2 = a$$

$$v_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, v_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

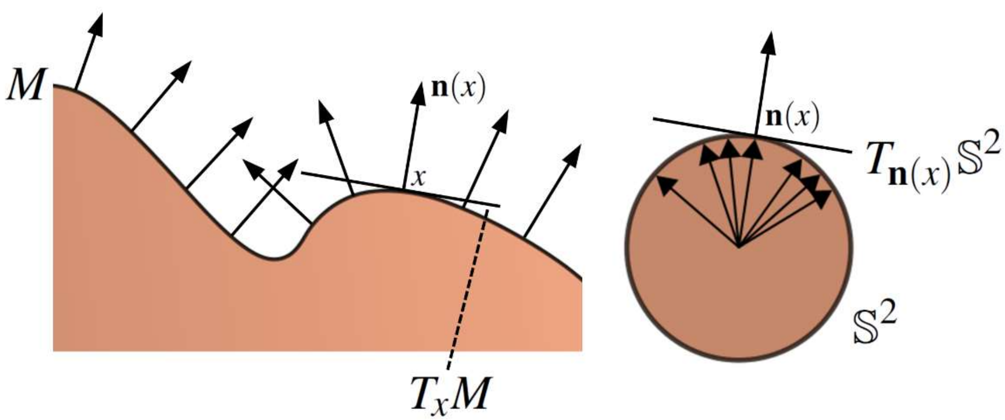



degenerate means determinant = 0 (at least one eigenvalue = 0);
bi-linear is simple: a = 0 means degenerated to
linear anyway: no critical point at all! (except constant function)
(but with more than one cell: can have max or min at vertices)

# Interlude: Curvature and Shape Operator

Gauss map

$$\mathbf{n}: M \to \mathbb{S}^2$$
$$x \mapsto \mathbf{n}(x)$$




Principal curvature magnitudes and directions are eigenvalues and eigenvectors of shape operator **S**

$$T_{\mathbf{n}(x)}\mathbb{S}^2 \cong T_xM$$

Differential of Gauss map

$$d\mathbf{n}: TM \to T\mathbb{S}^2$$
$$\mathbf{v} \mapsto d\mathbf{n}(\mathbf{v})$$

$$(d\mathbf{n})_x: T_xM \to T_{\mathbf{n}(x)}\mathbb{S}^2$$
$$\mathbf{v} \mapsto d\mathbf{n}(\mathbf{v})$$

*Shape operator (Weingarten map)*

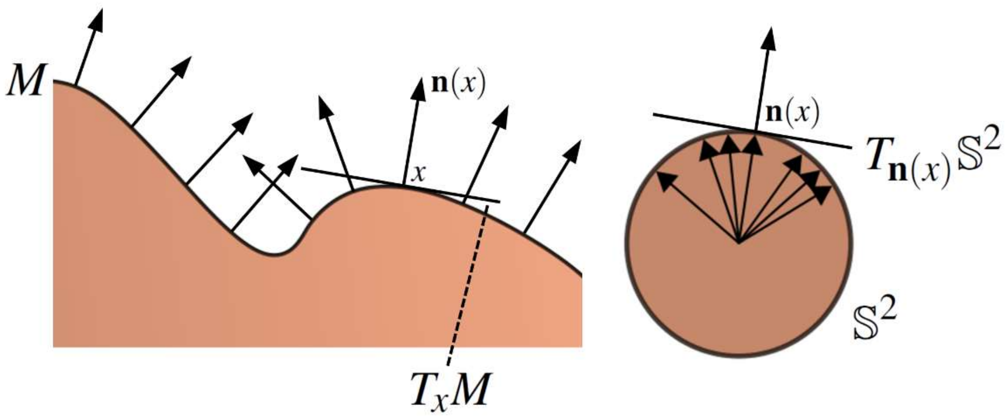
$$\mathbf{S}: TM \to TM$$

$$\mathbf{S}_x: T_xM \to T_xM$$
$$\mathbf{v} \mapsto \mathbf{S}_x(\mathbf{v}) = d\mathbf{n}(\mathbf{v})$$

# Interlude: Curvature and Shape Operator

## Gauss map

$$\mathbf{n}: M \to \mathbb{S}^2$$
$$x \mapsto \mathbf{n}(x)$$




Principal curvature magnitudes and directions are eigenvalues and eigenvectors of shape operator **S**

$$T_{\mathbf{n}(x)}\mathbb{S}^2 \cong T_xM$$

## Differential of Gauss map

$$d\mathbf{n}: TM \to T\mathbb{S}^2$$
$$\mathbf{v} \mapsto d\mathbf{n}(\mathbf{v})$$

$$(d\mathbf{n})_x: T_xM \to T_{\mathbf{n}(x)}\mathbb{S}^2$$
$$\mathbf{v} \mapsto d\mathbf{n}(\mathbf{v})$$

## *Shape operator (Weingarten map)*

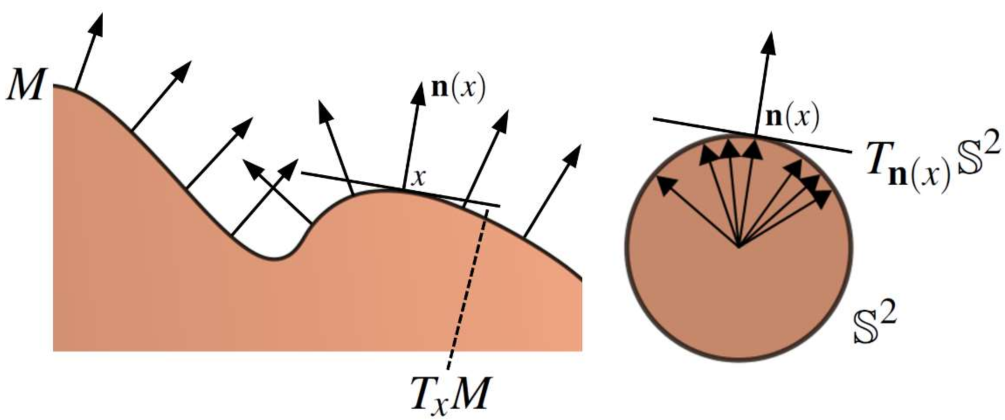
$$\mathbf{S}: TM \to TM$$

$$\mathbf{S}_x: T_xM \to T_xM$$
$$\mathbf{v} \mapsto \mathbf{S}_x(\mathbf{v}) = \nabla_{\mathbf{v}}\mathbf{n}$$

# Interlude: Curvature and Shape Operator

## Gauss map

$$\mathbf{n}: M \to \mathbb{S}^2$$
$$x \mapsto \mathbf{n}(x)$$




Principal curvature magnitudes and directions are eigenvalues and eigenvectors of shape operator **S**

$$T_{\mathbf{n}(x)}\mathbb{S}^2 \cong T_x M$$

## Differential of Gauss map

$$d\mathbf{n}: TM \to T\mathbb{S}^2$$
$$\mathbf{v} \mapsto d\mathbf{n}(\mathbf{v})$$

$$(d\mathbf{n})_x: T_x M \to T_{\mathbf{n}(x)}\mathbb{S}^2$$
$$\mathbf{v} \mapsto d\mathbf{n}(\mathbf{v})$$

## *Shape operator (Weingarten map)*

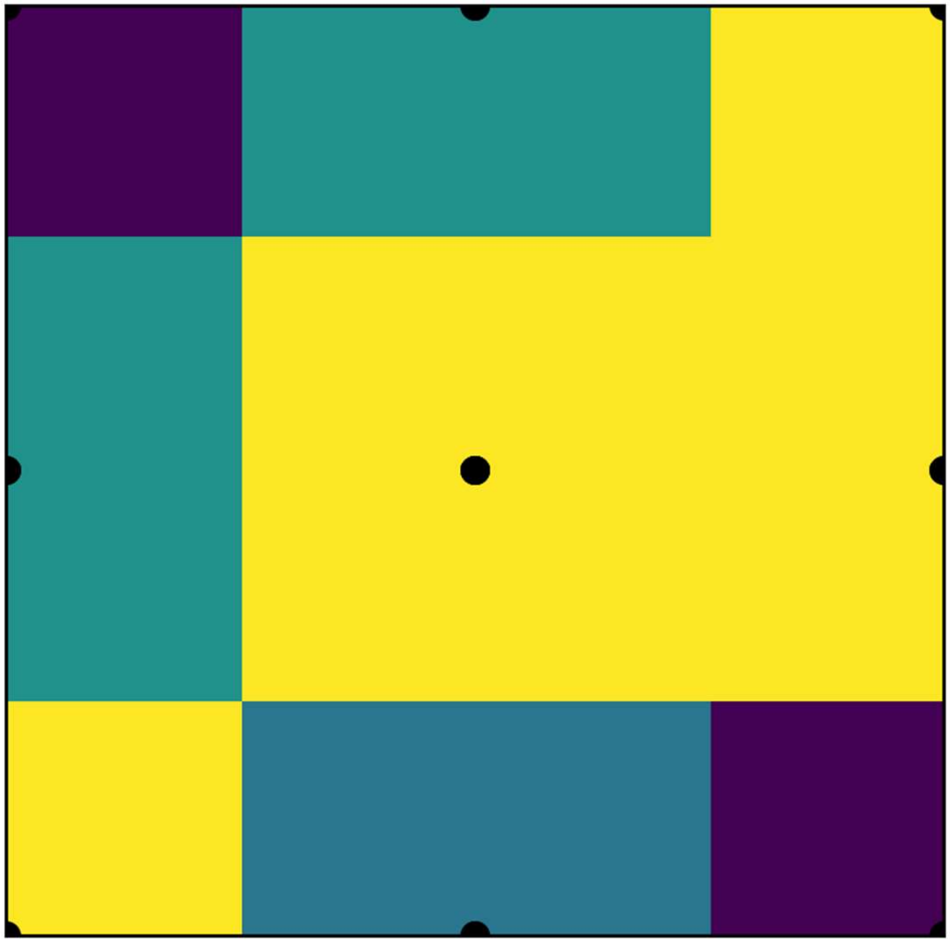
$$\mathbf{S}: TM \to TM$$

$$\mathbf{S}_x: T_x M \to T_x M$$
$$\mathbf{v} \mapsto \mathbf{S}_x(\mathbf{v}) = -\nabla_{\mathbf{v}}\mathbf{n}$$

(sign is convention)

# Bi-Linear Interpolation: Comparisons

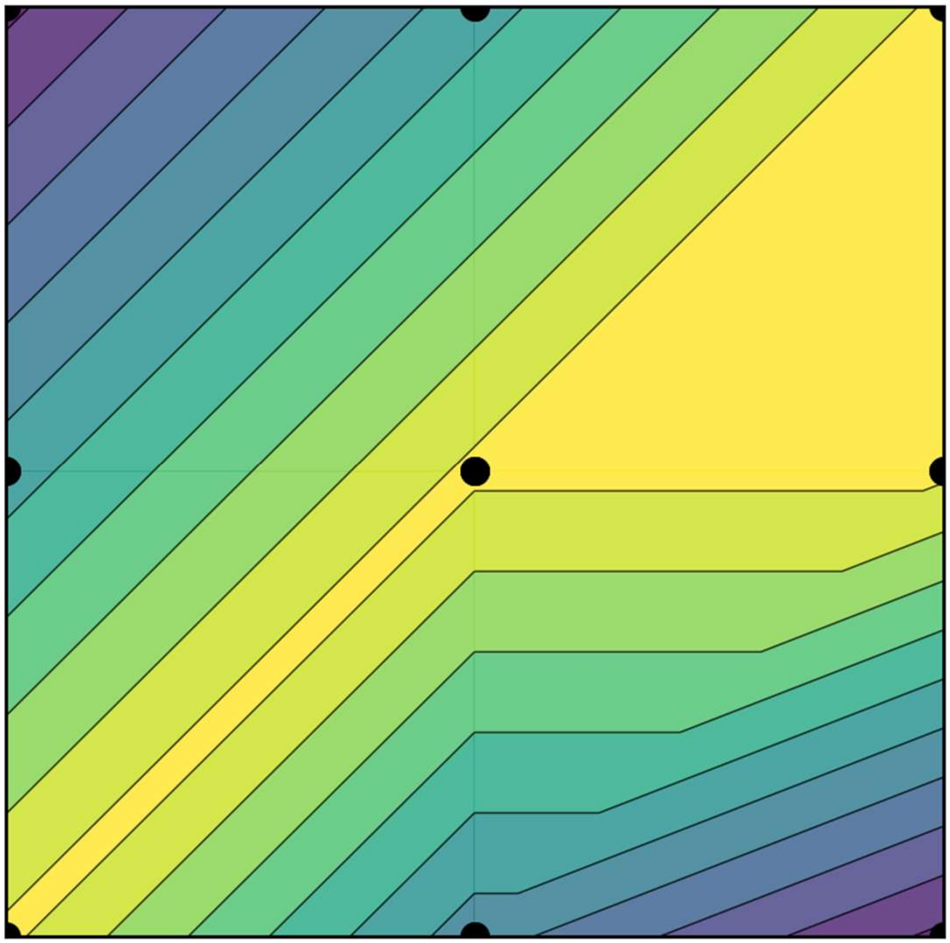
nearest-neighbor



# Bi-Linear Interpolation: Comparisons

linear

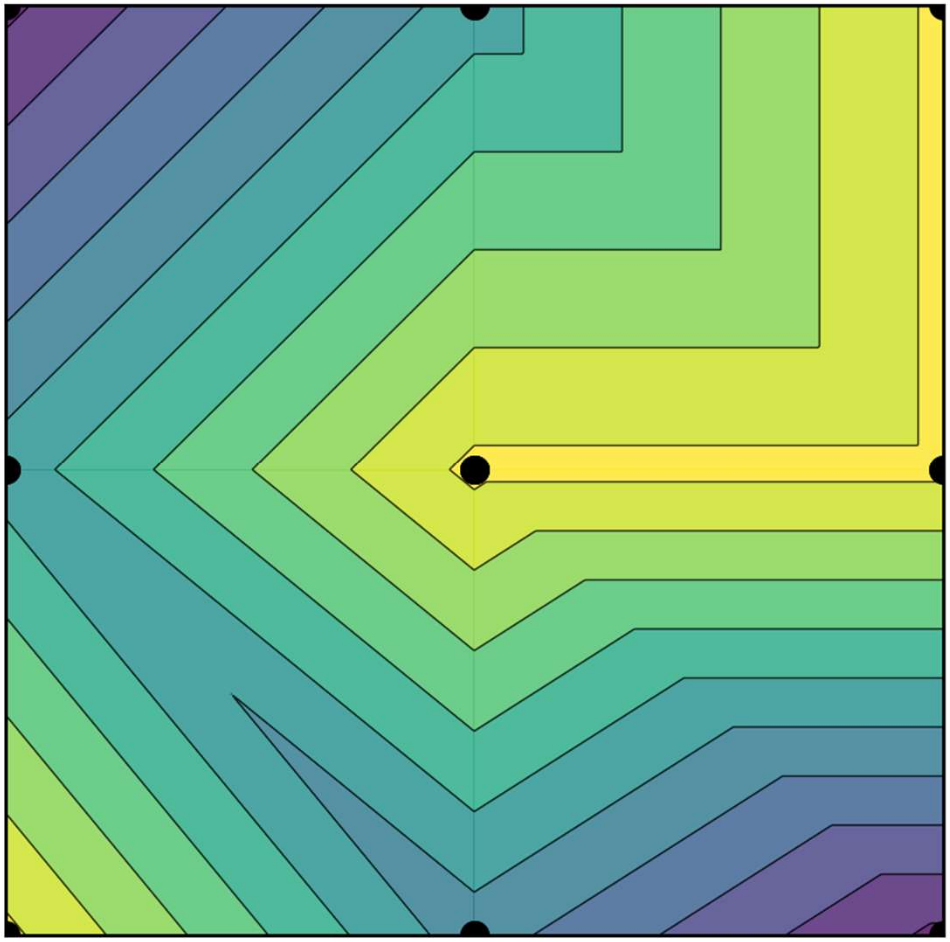
(2 triangles per quad;
diagonal:
bottom-left,
top-right)



# Bi-Linear Interpolation: Comparisons

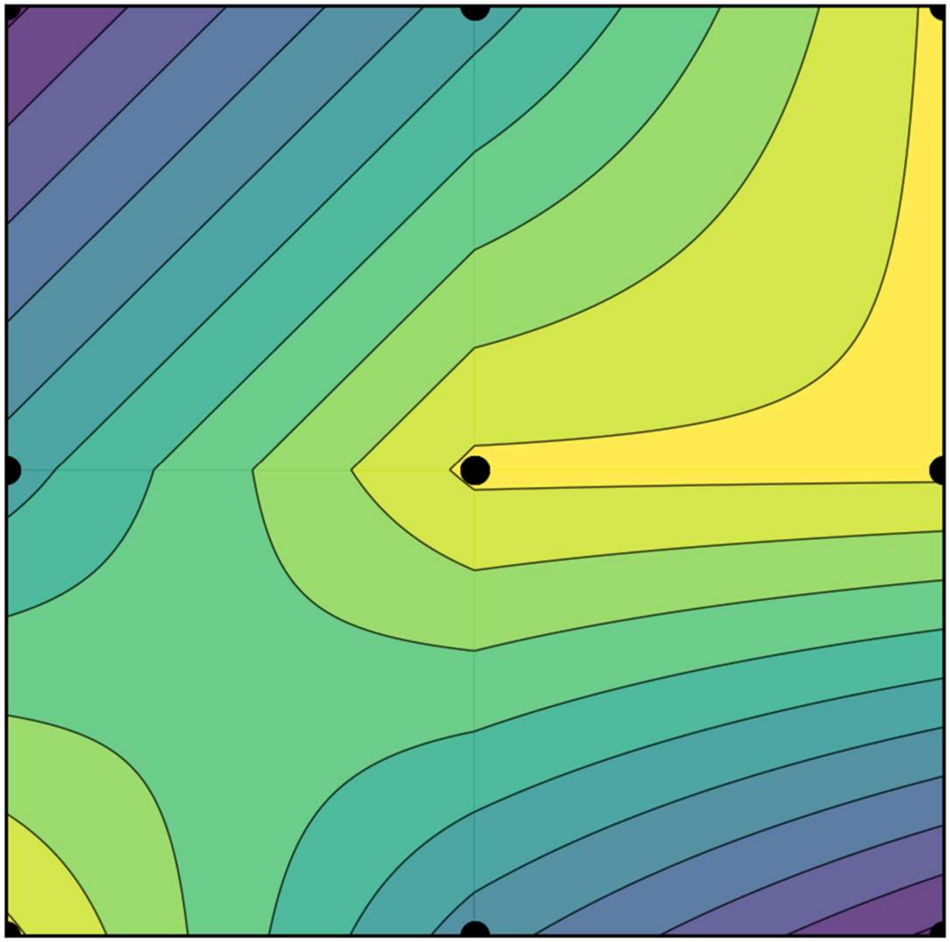
linear

(2 triangles per quad;
diagonal:
top-left,
bottom-right)



# Bi-Linear Interpolation: Comparisons

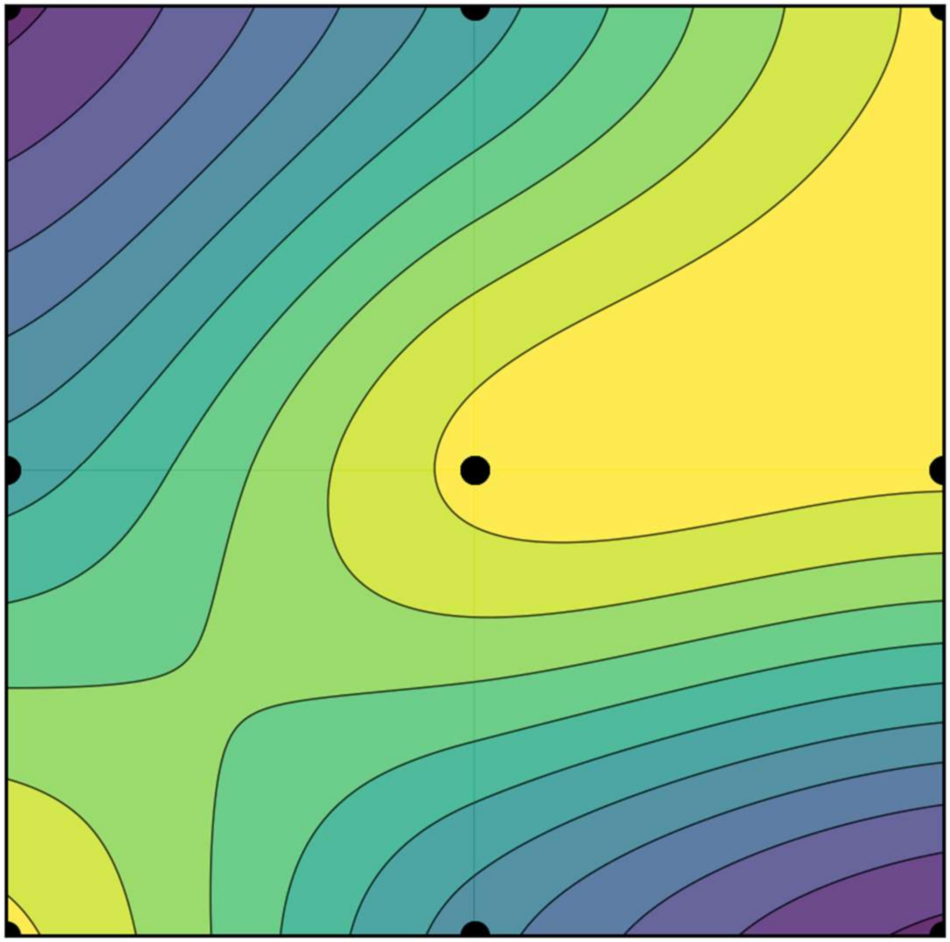
bi-linear



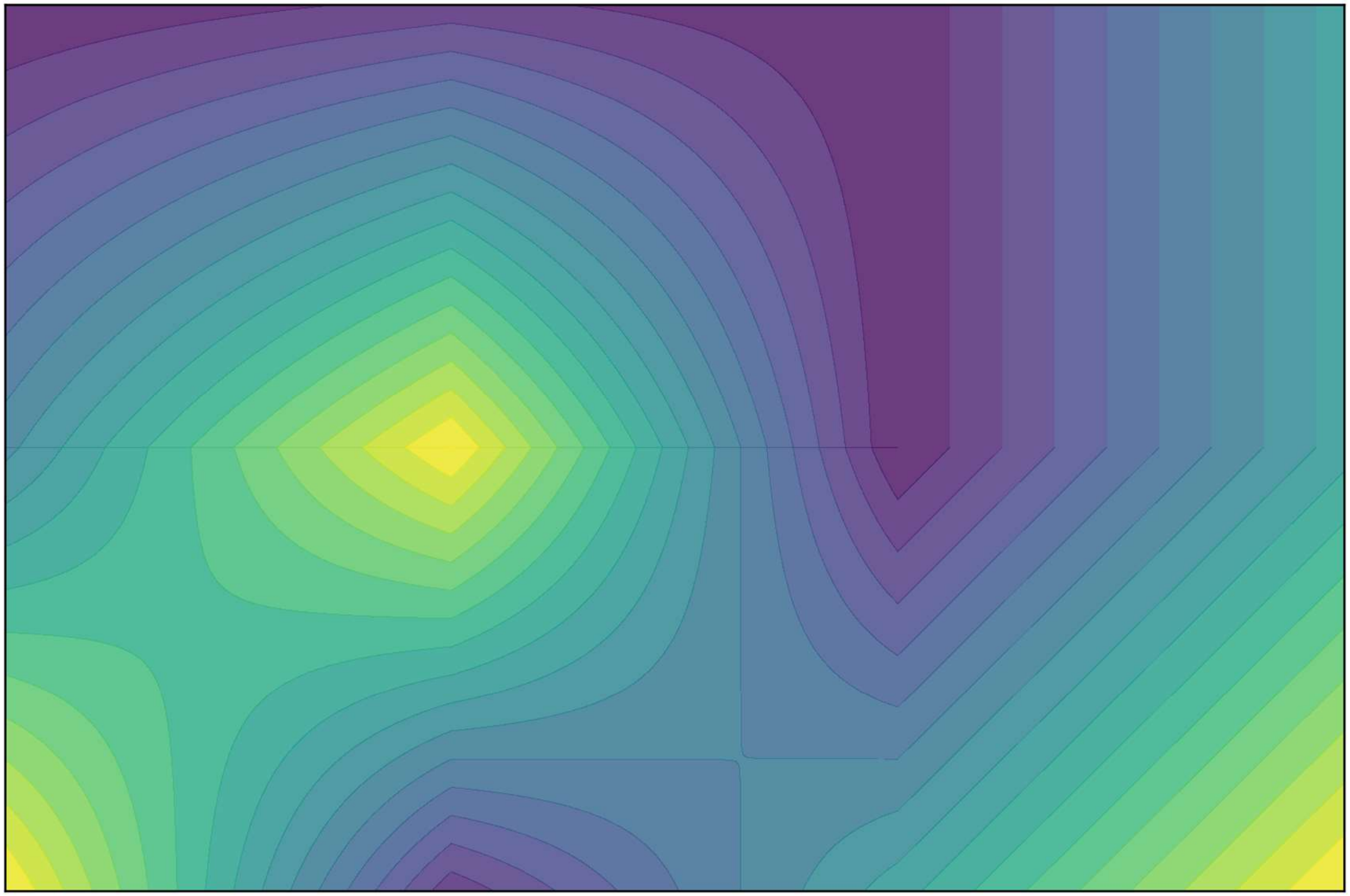
# Bi-Linear Interpolation: Comparisons

bi-cubic

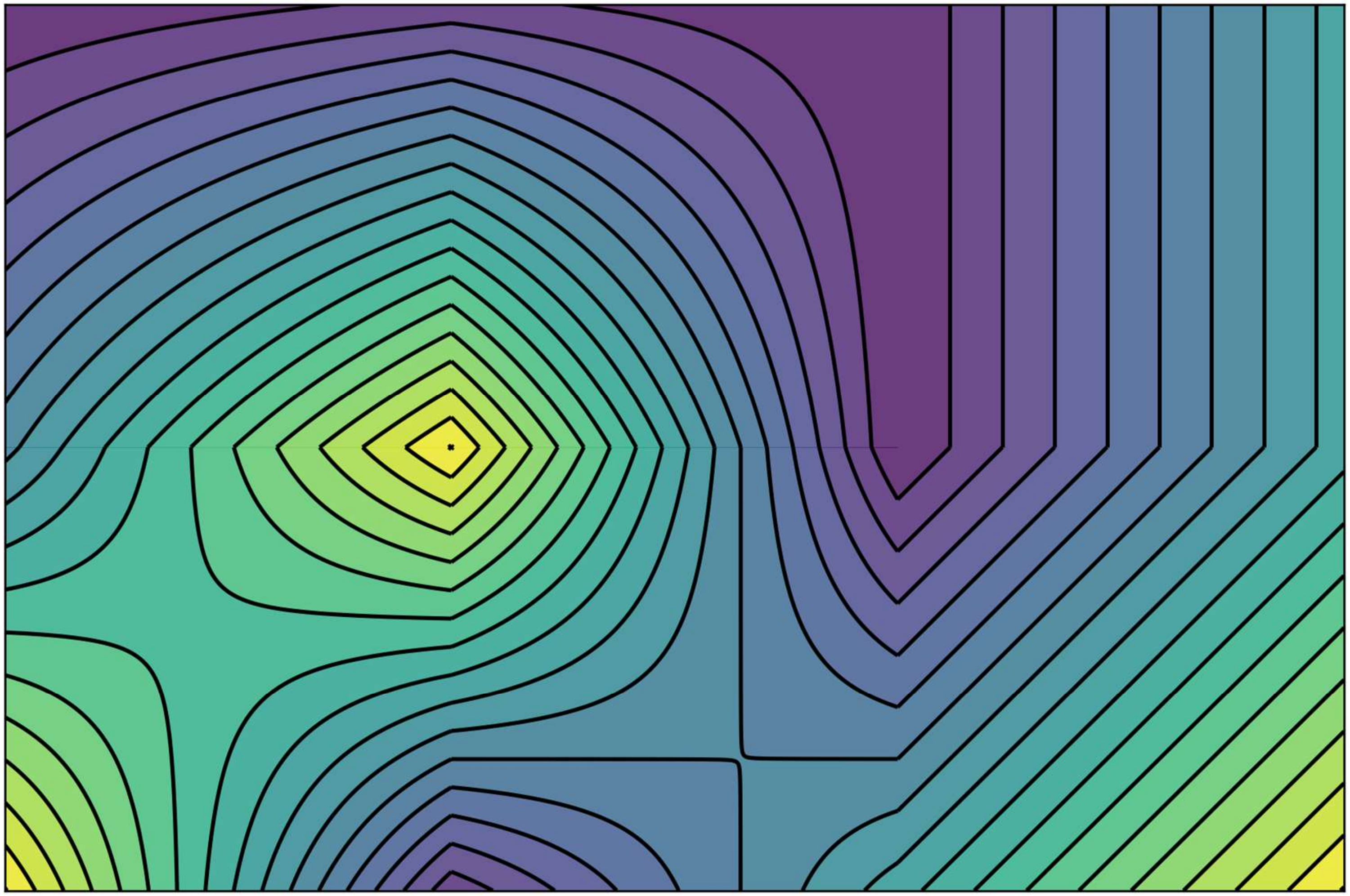
(Catmull-Rom spline)



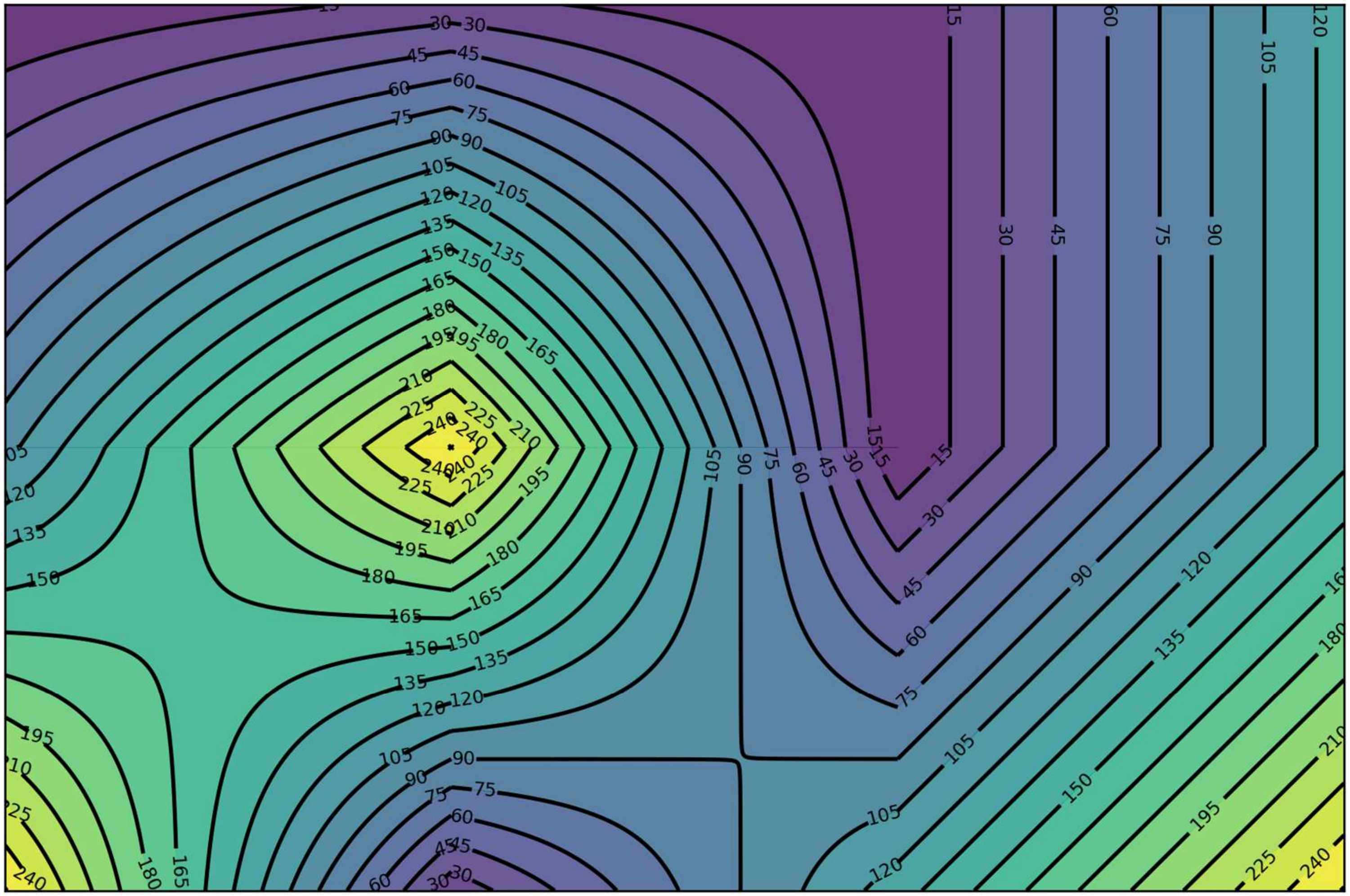
# Piecewise Bi-Linear (Example: 3x2 Cells)



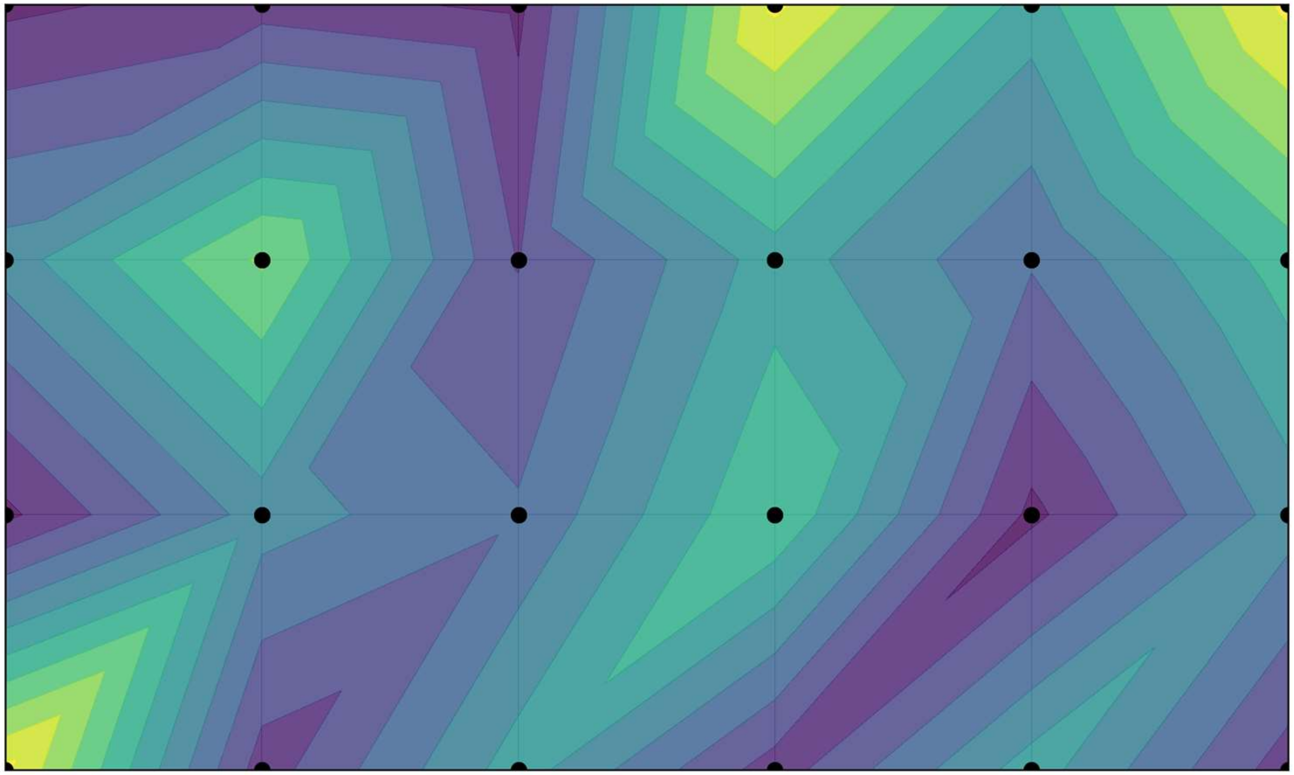
# Piecewise Bi-Linear (Example: 3x2 Cells)



# Piecewise Bi-Linear (Example: 3x2 Cells)

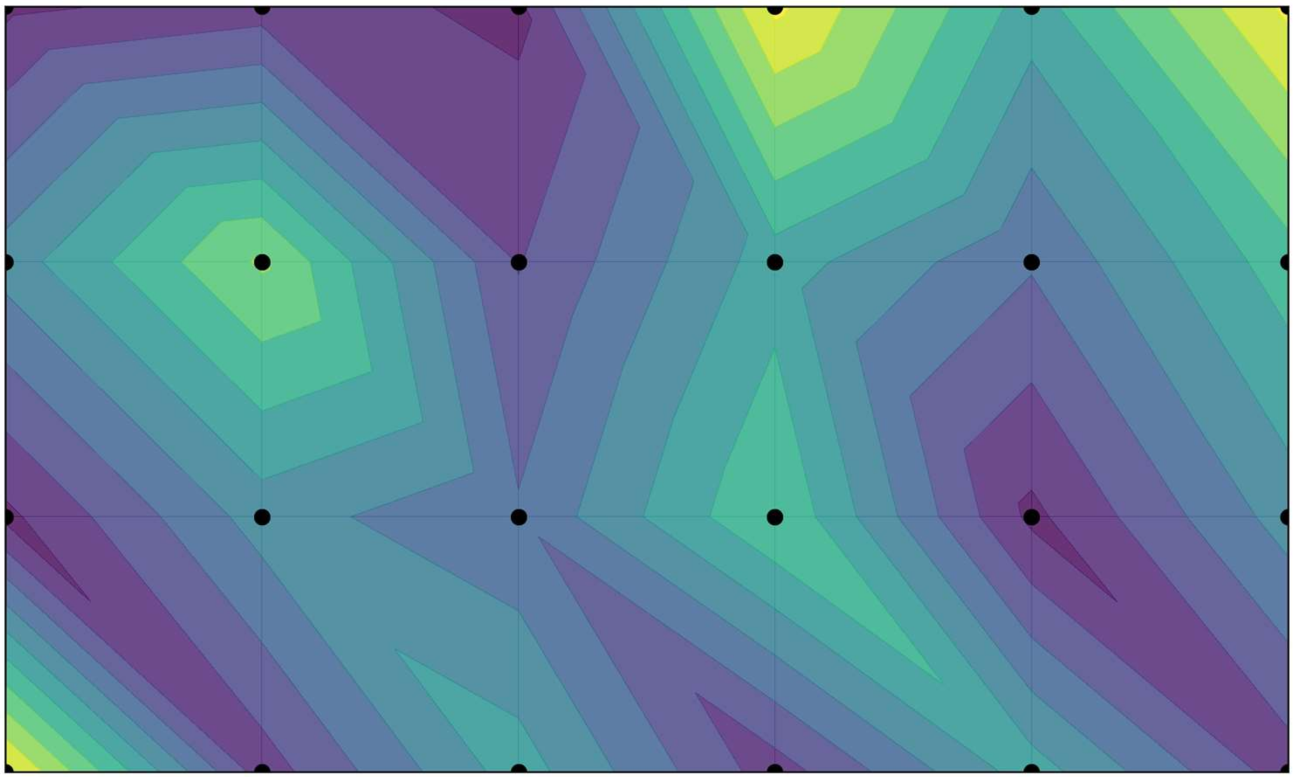


# Bi-Linear Interpolation: Comparisons



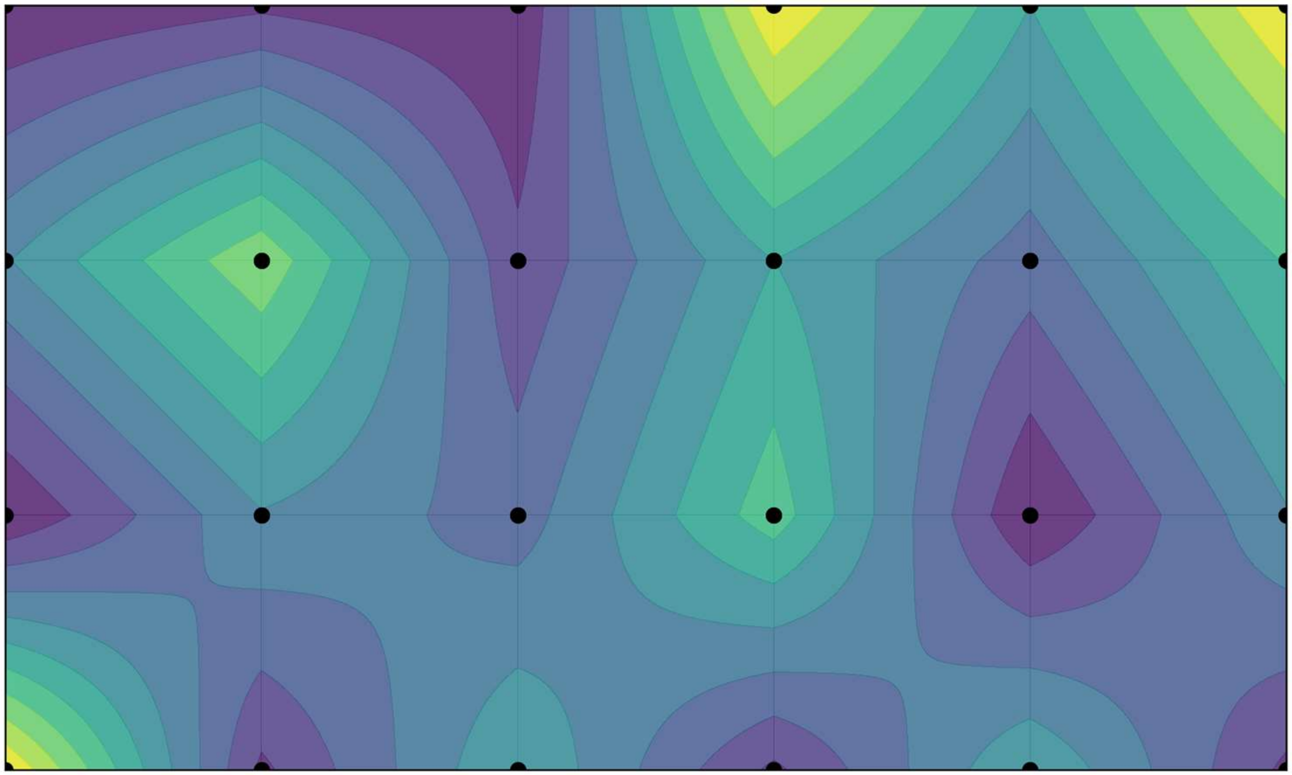
linear (diagonal 1)

# Bi-Linear Interpolation: Comparisons



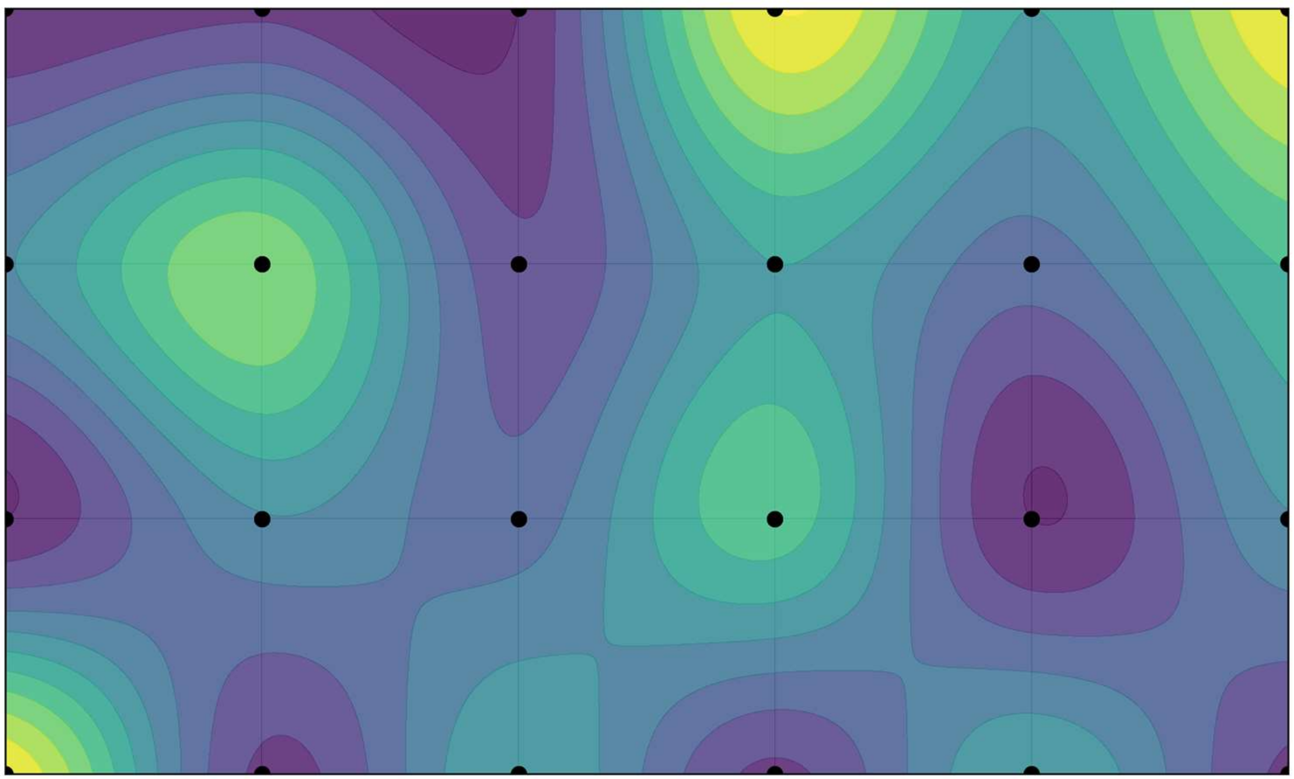
linear (diagonal 2)

# Bi-Linear Interpolation: Comparisons



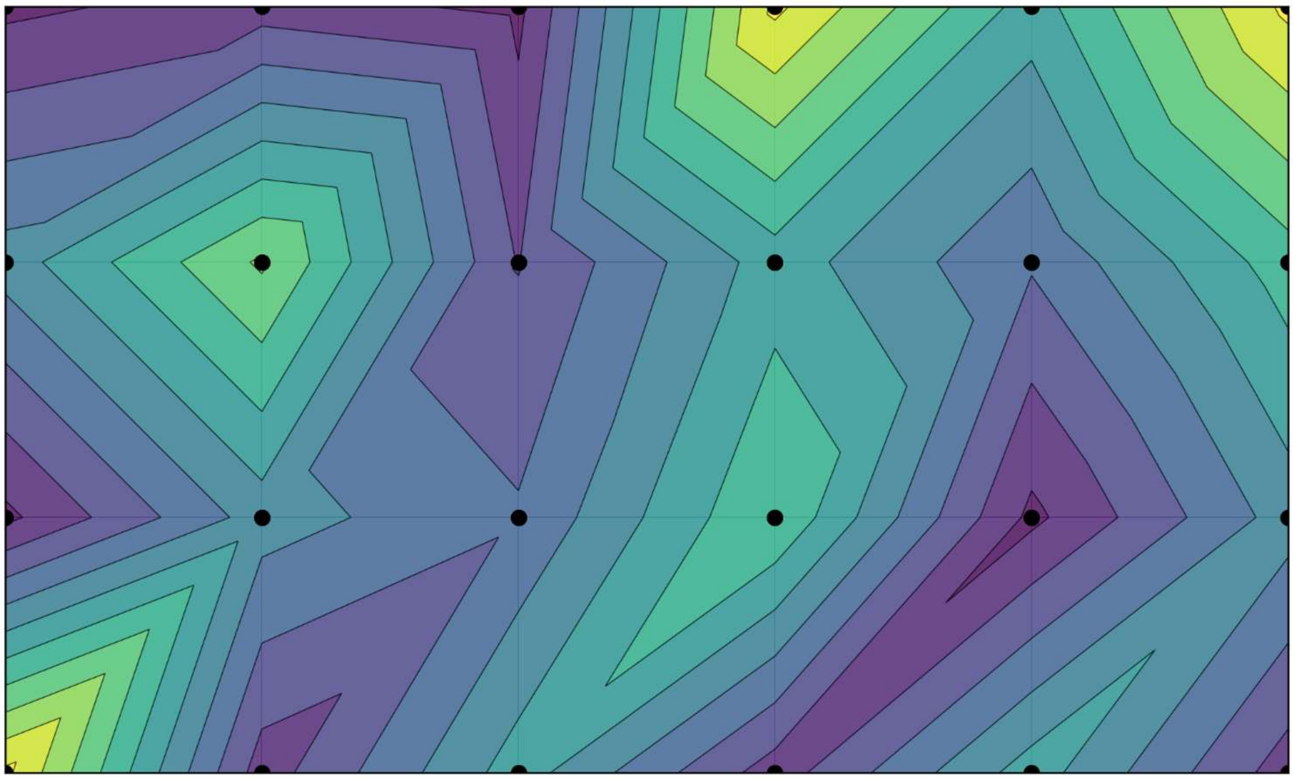
bi-linear (in 3D: tri-linear)

# Bi-Linear Interpolation: Comparisons



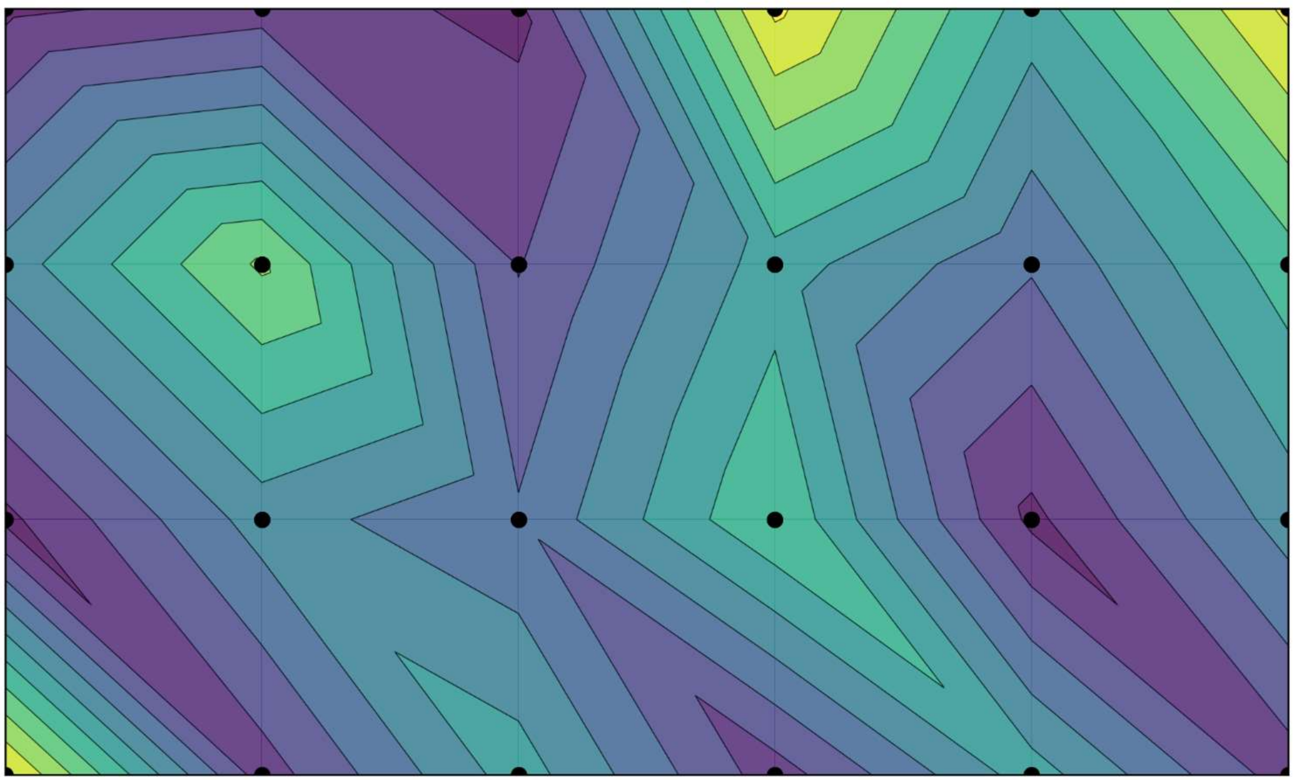
bi-cubic (in 3D: tri-cubic)

# Bi-Linear Interpolation: Comparisons



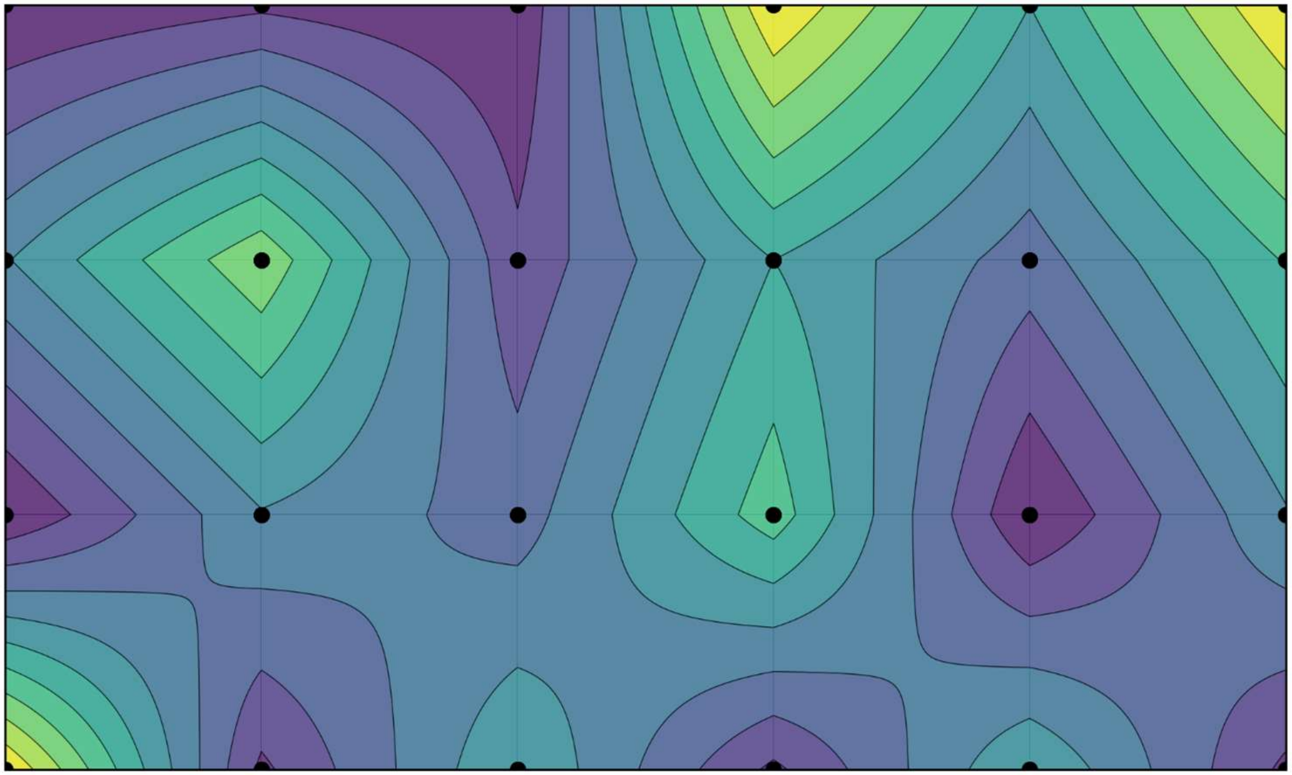
linear (diagonal 1)

# Bi-Linear Interpolation: Comparisons



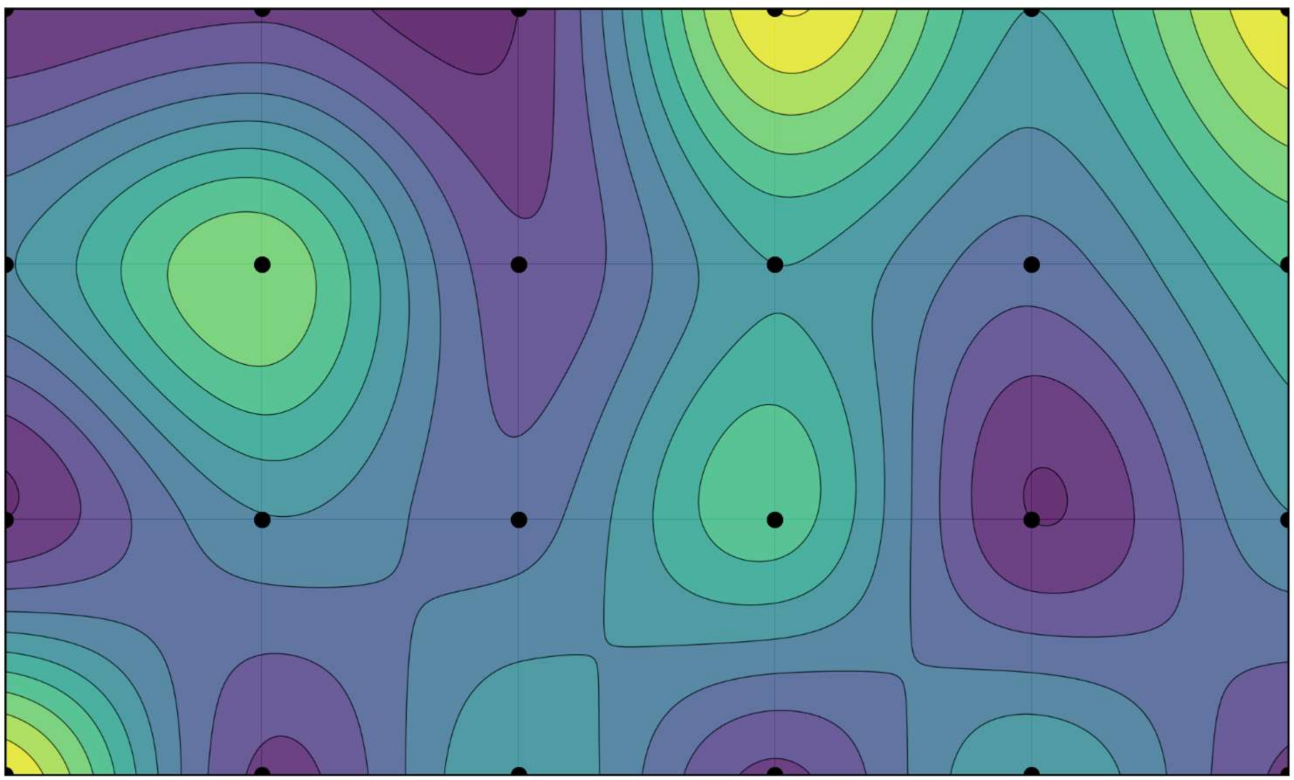
linear (diagonal 2)

# Bi-Linear Interpolation: Comparisons



bi-linear (in 3D: tri-linear)

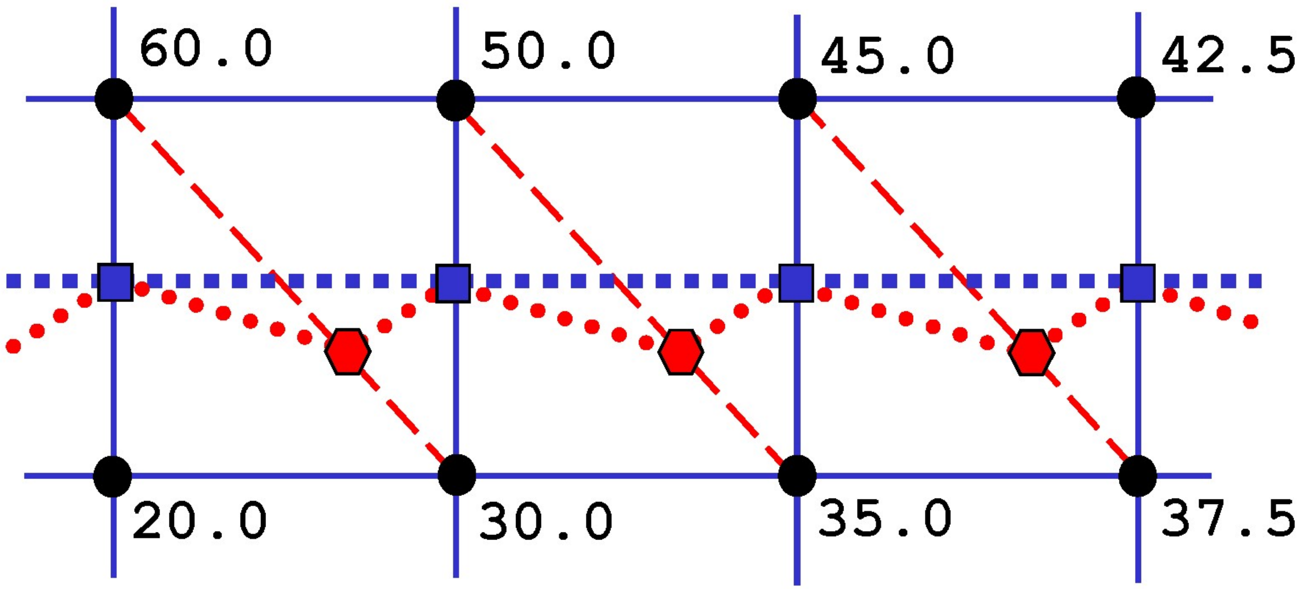
# Bi-Linear Interpolation: Comparisons



bi-cubic (in 3D: tri-cubic)

## *Contours in triangle/tetrahedral cells*

Illustrative example: Find contour at level *c*=40.0 !




original quad grid, yielding vertices ■ and contour ▪▪▪▪▪

triangulated grid, yielding vertices ⬢ and contour •••••

# Thank you.

Thanks for material

- Helwig Hauser
- Eduard Gröller
- Daniel Weiskopf
- Torsten Möller
- Ronny Peikert
- Philipp Muigg
- Christof Rezk-Salama