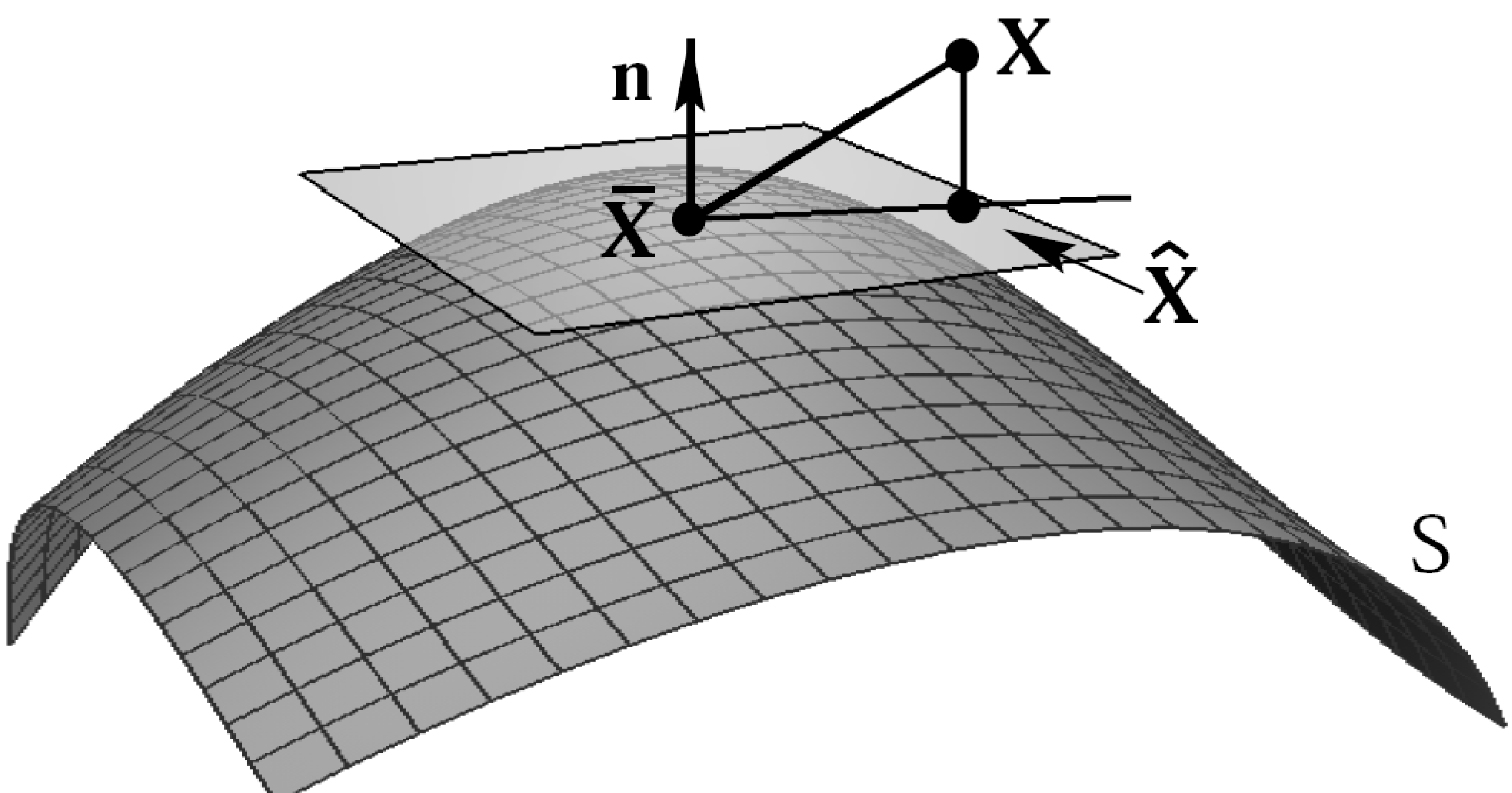
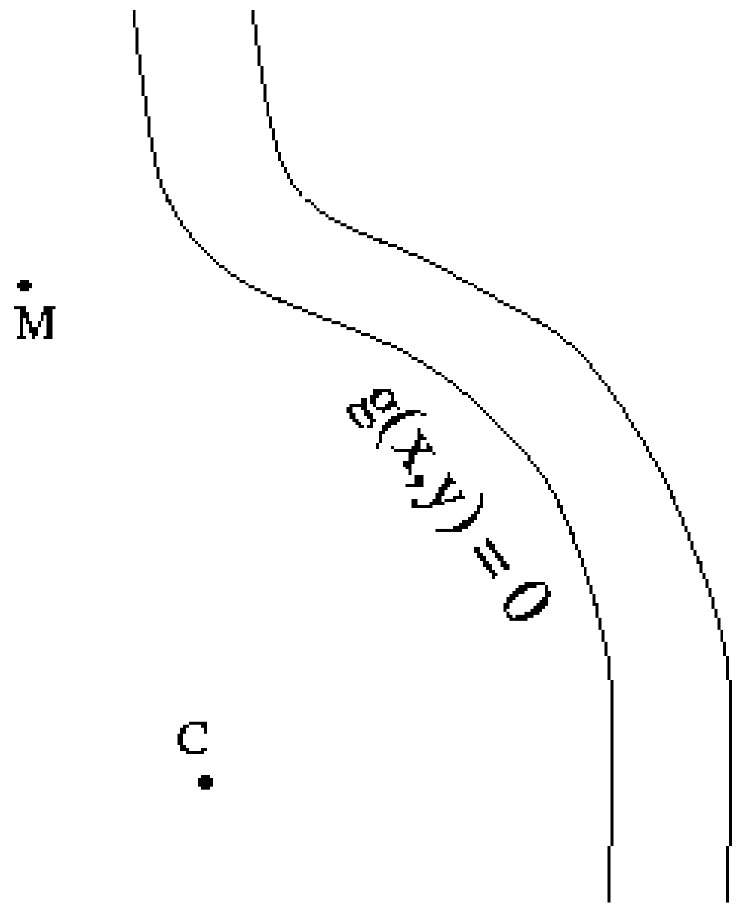
# Cpt 5: Algorithm Evaluation and Error Analysis



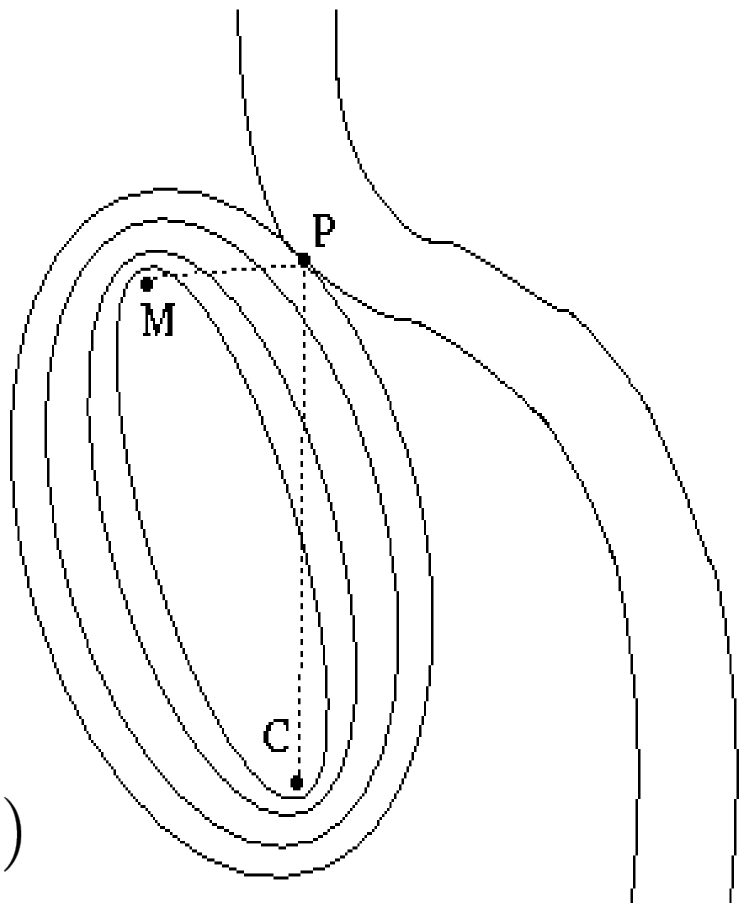
# Preliminary: Lagrange Multipliers

- Milkmaid problem



# Preliminary: Lagrange Multipliers

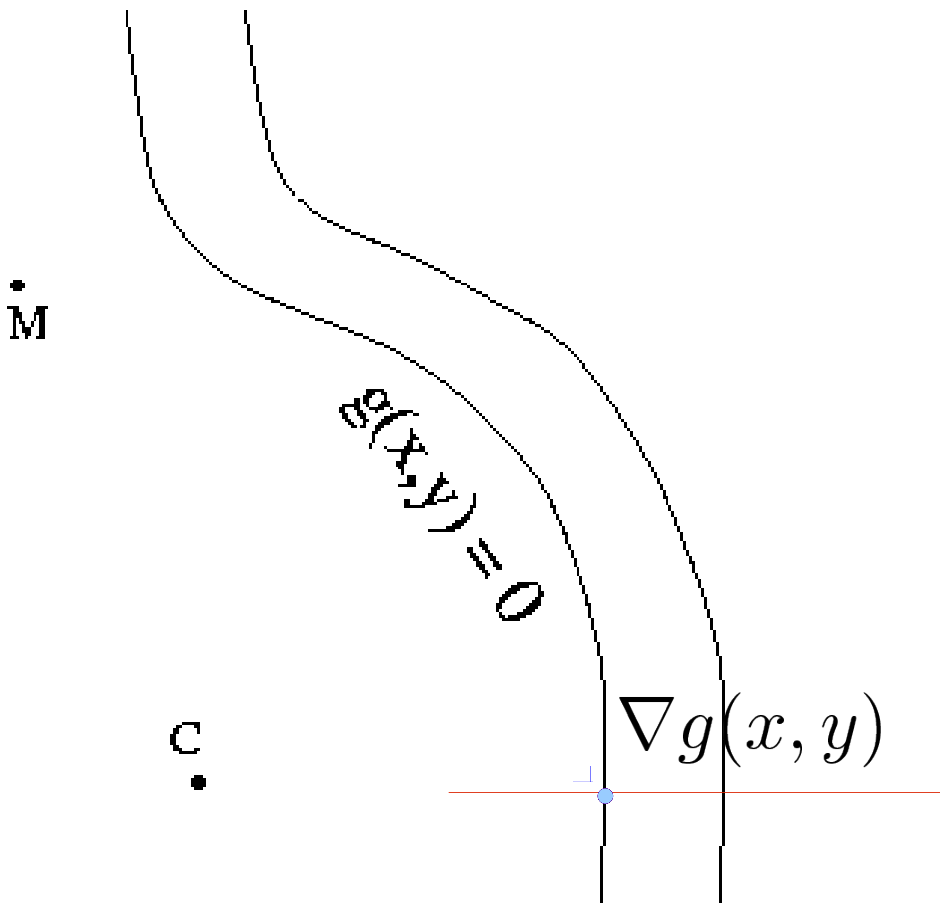
- Milkmaid problem




$$f(\mathbf{x}) = (MP + CP)$$

# Preliminary: Lagrange Multipliers

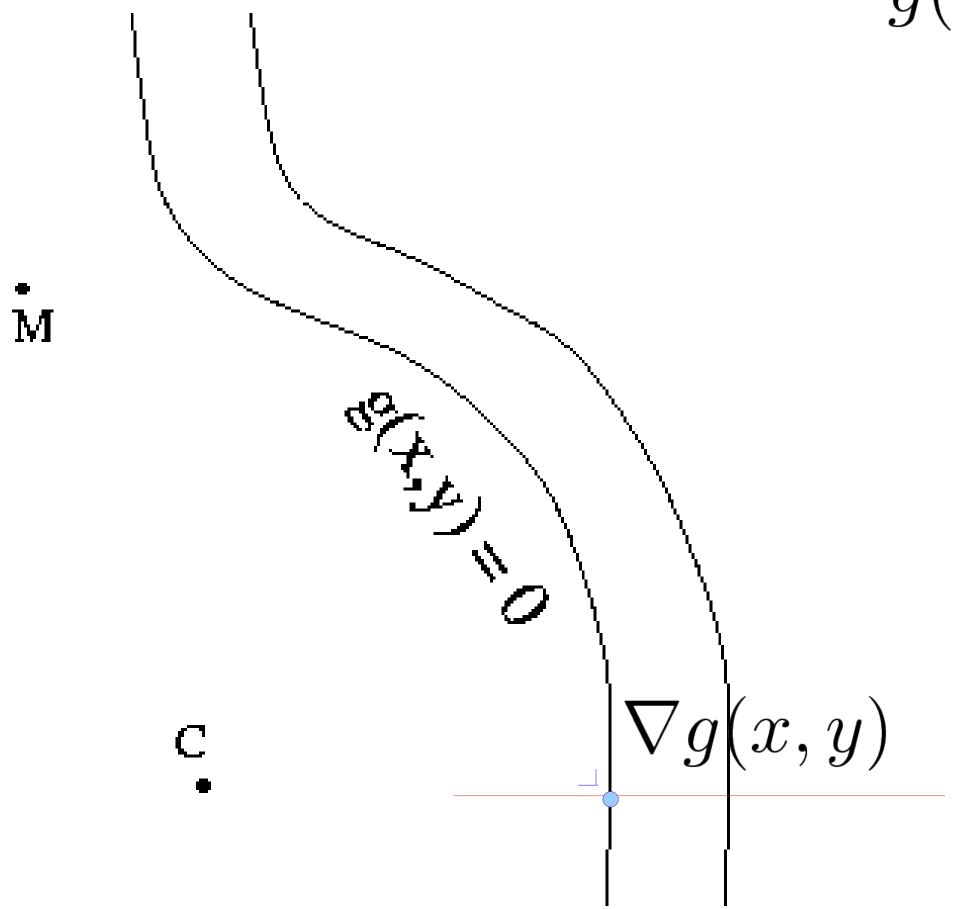
- Milkmaid problem



# Preliminary: Lagrange Multipliers

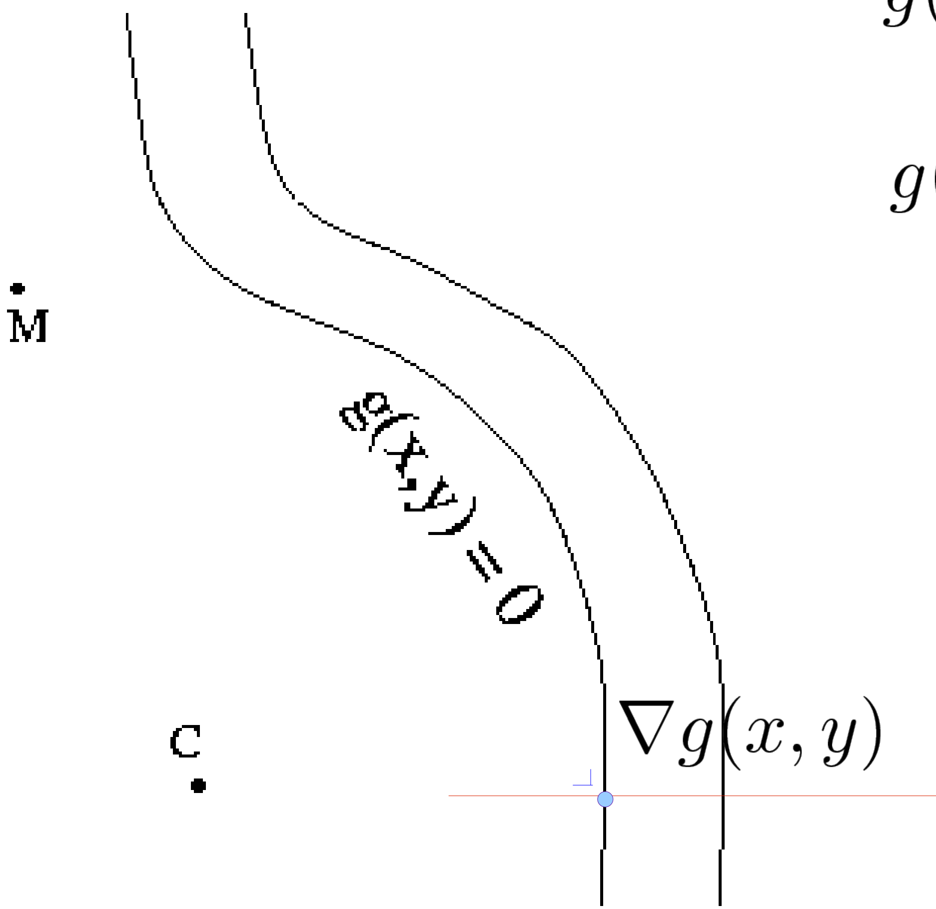
- Milkmaid problem

$$g(\mathbf{x}+\epsilon) \approx g(x) + \epsilon^T \nabla g(\mathbf{x})$$



# Preliminary: Lagrange Multipliers

- Milkmaid problem

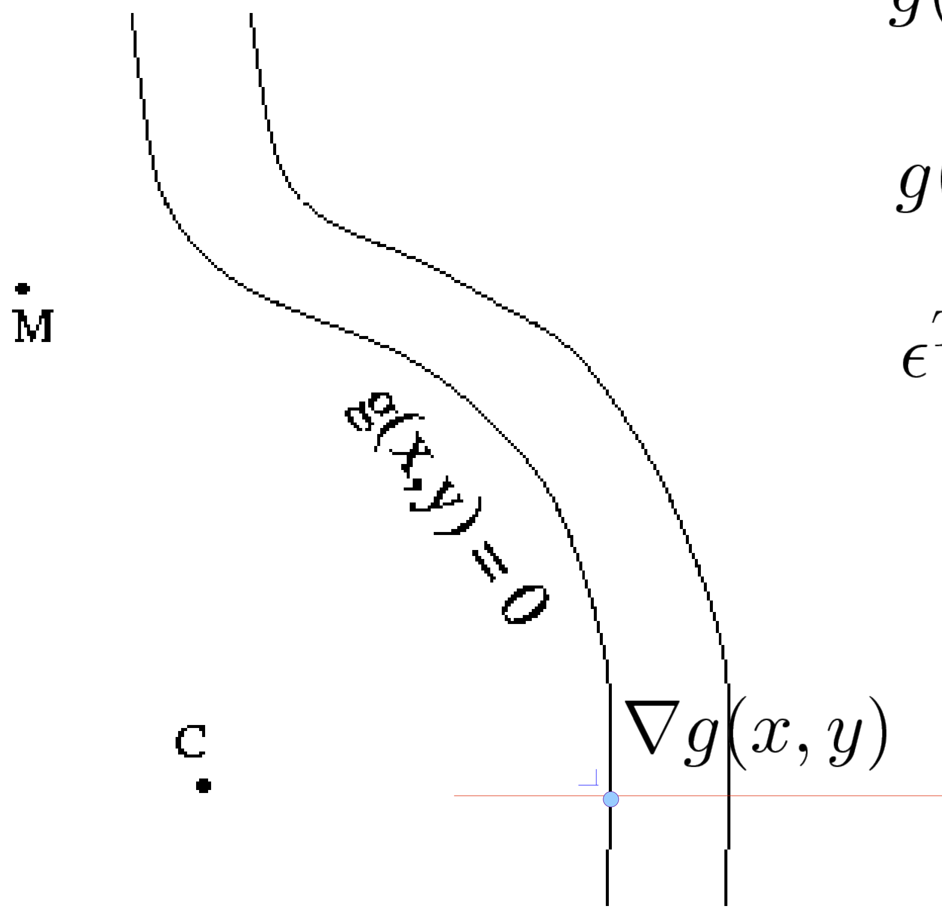



$$g(\mathbf{x} + \epsilon) \approx g(x) + \epsilon^T \nabla g(\mathbf{x})$$

$$g(\mathbf{x} + \epsilon) = g(\mathbf{x}), \epsilon \to 0$$

# Preliminary: Lagrange Multipliers

- Milkmaid problem



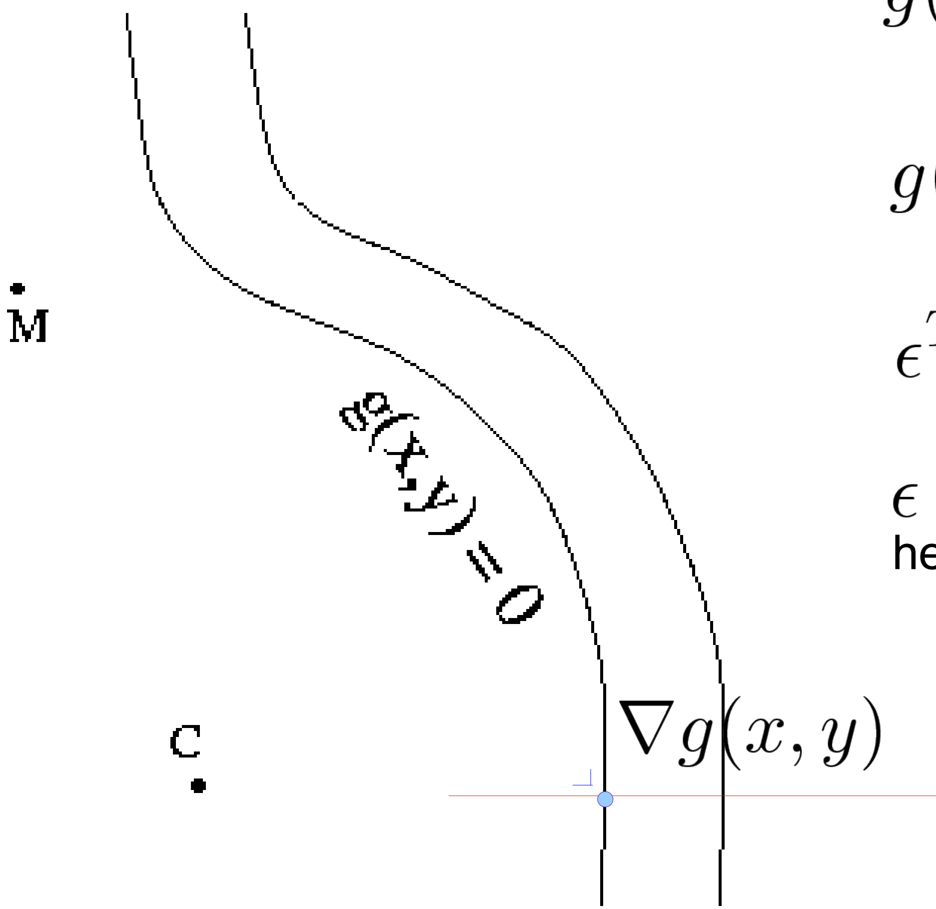

$$g(\mathbf{x} + \epsilon) \approx g(x) + \epsilon^T \nabla g(\mathbf{x})$$

$$g(\mathbf{x} + \epsilon) = g(\mathbf{x}), \epsilon \rightarrow 0$$

$$\epsilon^T \nabla g(\mathbf{x}) = 0$$

# Preliminary: Lagrange Multipliers

- Milkmaid problem




$$g(\mathbf{x} + \epsilon) \approx g(x) + \epsilon^T \nabla g(\mathbf{x})$$

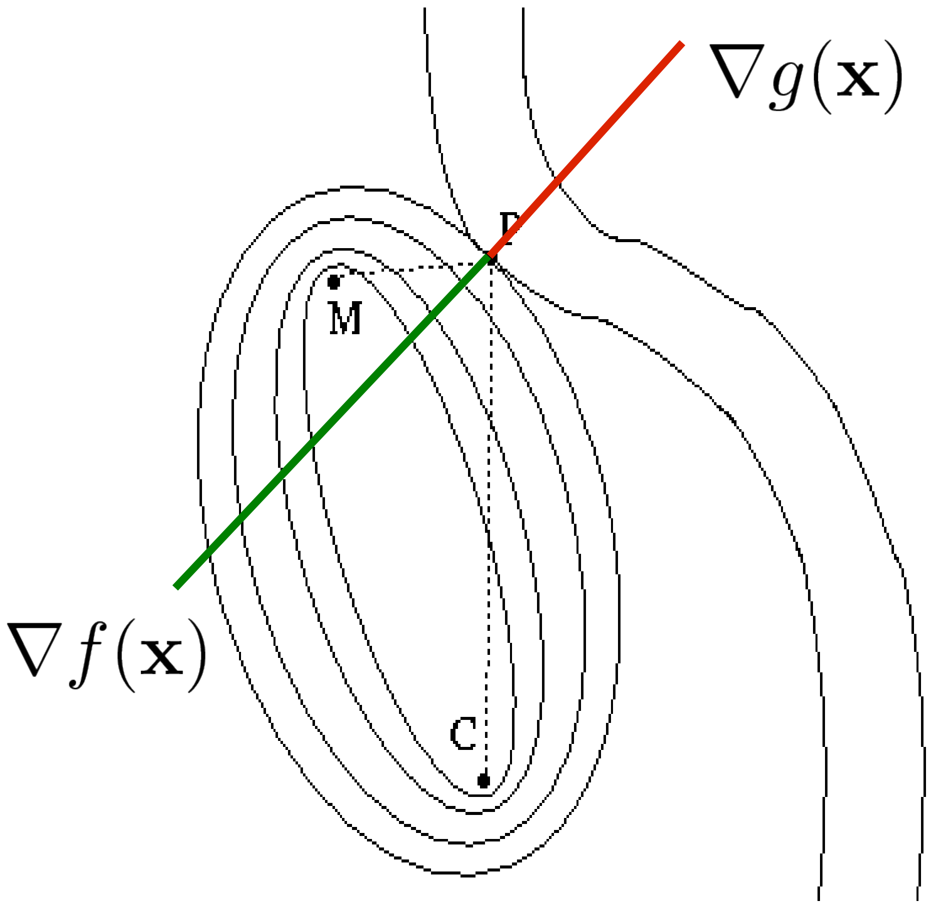
$$g(\mathbf{x} + \epsilon) = g(\mathbf{x}), \epsilon \to 0$$

$$\epsilon^T \nabla g(\mathbf{x}) = 0$$

$\epsilon$ is parallel to the surface of the function, hence $\nabla g(\mathbf{x})$ is orthogonal.

# Preliminary: Lagrange Multipliers

- Milkmaid problem



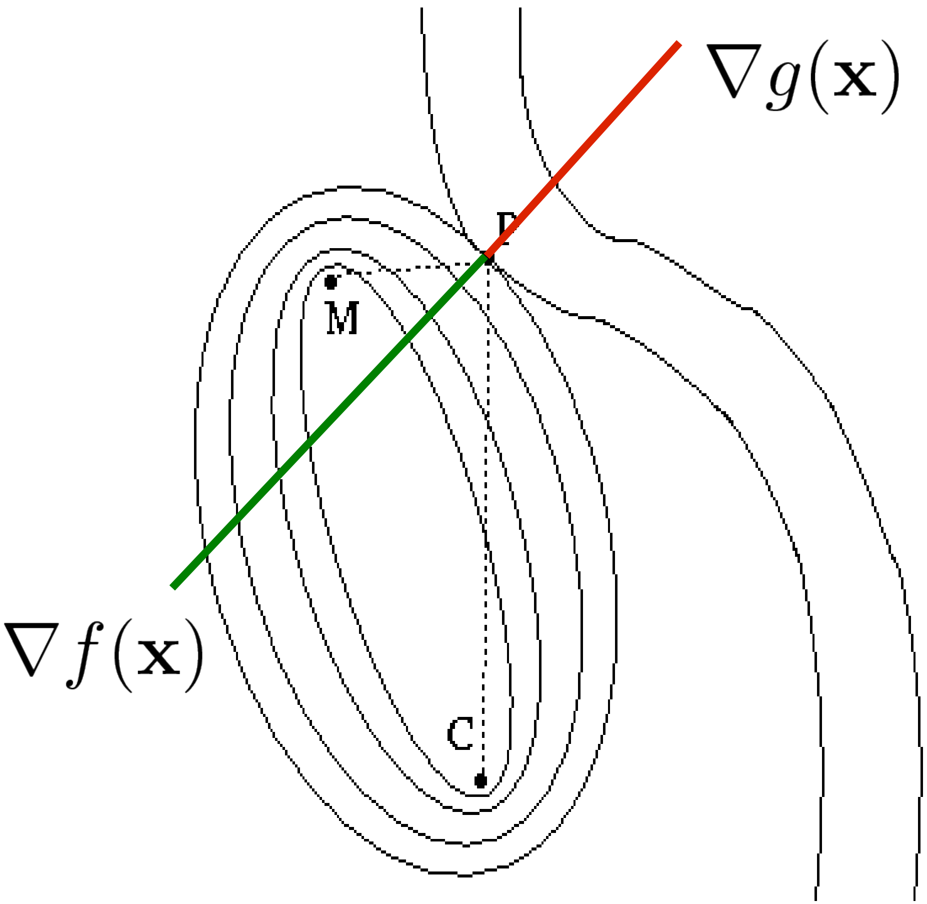

We now can calculate

$$\nabla f(\mathbf{x}) = \lambda \nabla g(\mathbf{x}),$$

where $\lambda$ is the ratio between the strength of the gradients and hence unimportant.

# Preliminary: Lagrange Multipliers

- Milkmaid problem

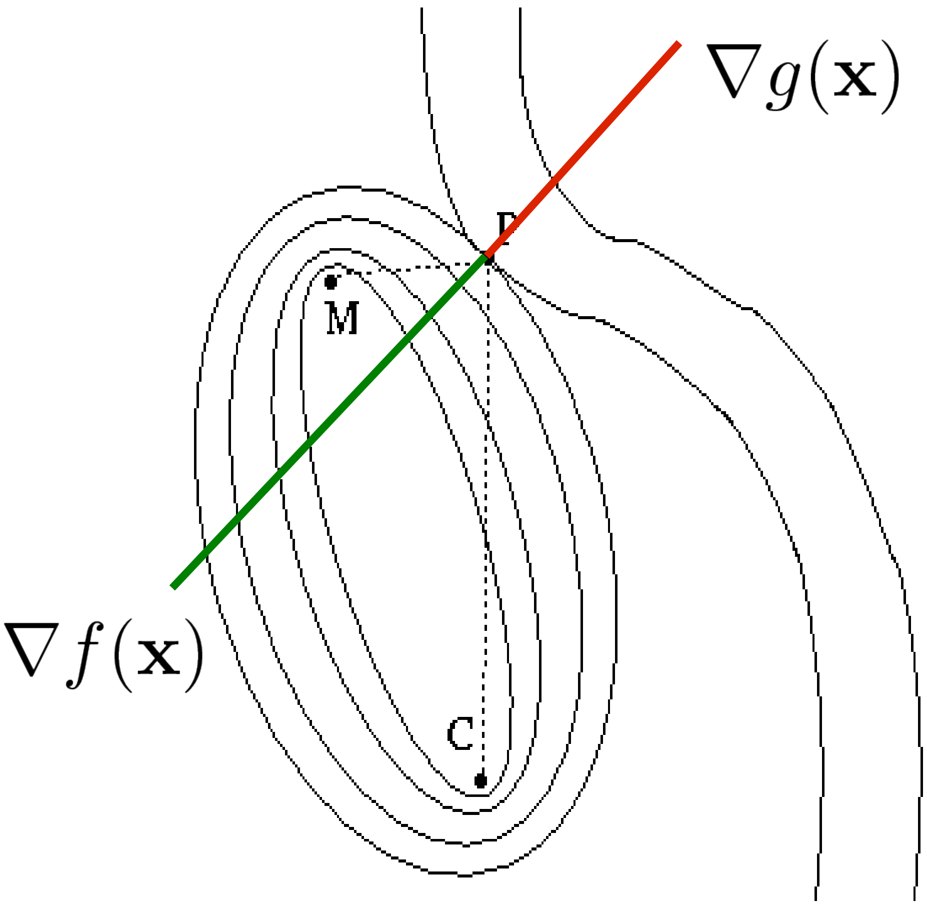



For convenience we introduce the *Lagrangian,* defined as

$$L(\mathbf{x}, \lambda) \equiv f(\mathbf{x}) + \lambda g(\mathbf{x})$$

# Preliminary: Lagrange Multipliers

- Milkmaid problem




For convenience we introduce the *Lagrangian,* defined as

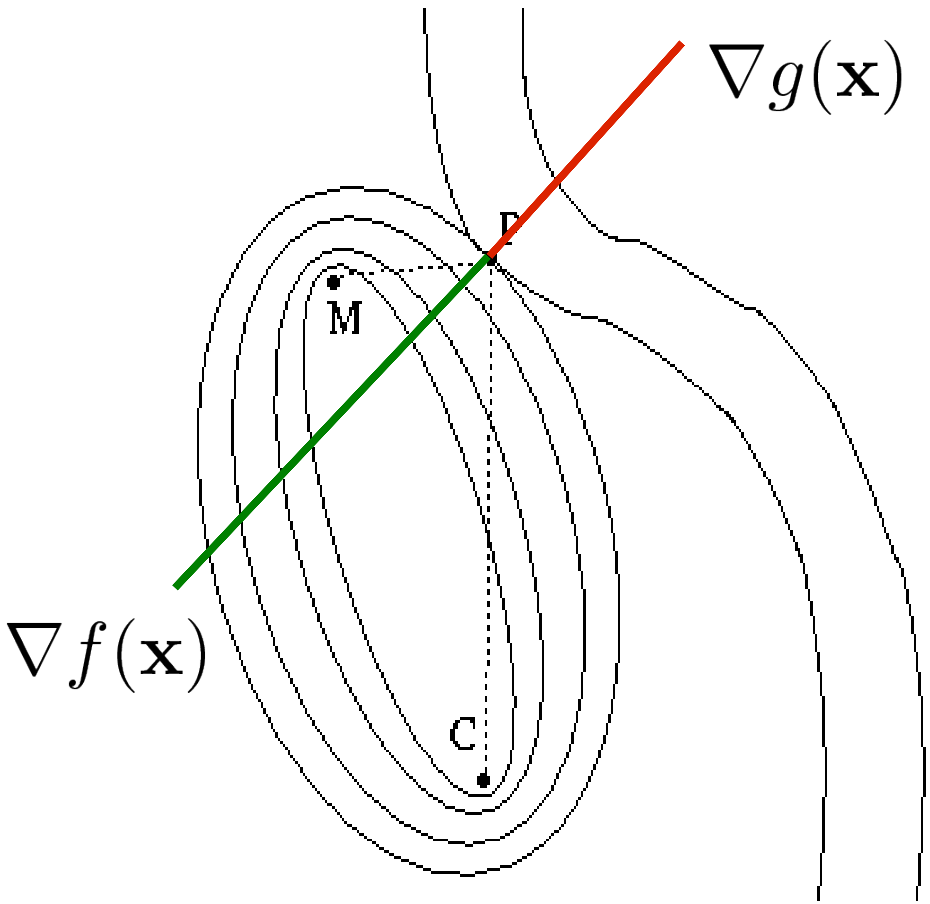
$$L(\mathbf{x}, \lambda) \equiv f(\mathbf{x}) + \lambda g(\mathbf{x})$$

Now, $\nabla_{\mathbf{x}} L = 0$ gives

$$\nabla f(\mathbf{x}) = \lambda \nabla g(\mathbf{x}),$$

# Preliminary: Lagrange Multipliers

- Milkmaid problem




For convenience we introduce the *Lagrangian,* defined as

$$L(\mathbf{x}, \lambda) \equiv f(\mathbf{x}) + \lambda g(\mathbf{x})$$

Now, $\nabla_{\mathbf{x}} L = 0$ gives

$$\nabla f(\mathbf{x}) = \lambda \nabla g(\mathbf{x}),$$

$\nabla_{\lambda} L = 0$ gives the constraint equation

$$g(\mathbf{x}) = 0$$

# Algorithm evaluation and Error Analysis

- Notation
- Let:

$\mathbf{x}$ be a measured image point

$\bar{\mathbf{x}}$ be a true image point

$\hat{\mathbf{x}}$ be an estimated image point

# Testing using synthetic data

- Create synthetic data between two images

$$\bar{\mathbf{x}}_{\mathbf{i}} \leftrightarrow \bar{\mathbf{x}}_{\mathbf{i}}{}'$$

- Corresponding points will be created w.r.t. a fixed transformation $\bar{H}$ :

$$\bar{\mathbf{x}}_{\mathbf{i}}{}' = \bar{H}\bar{\mathbf{x}}_{\mathbf{i}}$$

- Artificial Gaussian noise is added to these corresponding points, resulting in $\mathrm{x_i}$ and $\mathrm{x_i}'$
- Repeat may times with different amount of noise.

# Example: error in one image

- Noise is added to the second image only. Thus:

$$\mathbf{x_i} = \bar{\mathbf{x_i}}, \forall i$$

- Let $\mathbf{x_i} \leftrightarrow \mathbf{x_i}'$ be a set of noisy matched points between the images using Gaussian noise with variance $\sigma^2$ .
- The residual error is the average distance between the noisy input data $\mathbf{x_i}'$ and the estimated points $\hat{\mathbf{x}}_{\mathbf{i}}' = \hat{H}\bar{\mathbf{x_i}}$ :

$$\epsilon_{\text{res}} = \sqrt{\frac{1}{2n}\sum_{i=1}^{n} d(\mathbf{x_i}', \hat{\mathbf{x}}_{\mathbf{i}}')^2}$$

# Example: error in one image

- The value of the residual error is not in itself an absolute measure of the quality:
  - 4 matched points will be mapped exactly, resulting in a residual error of 0.
    - This is because $\hat{H}$ matches the projected points to the input data $\mathbf{x_i}'$, and not to the original data $\bar{\mathbf{x}}_\mathbf{i}'$.
    - Thus 4 points will match input data exactly, but does not give a very close approximation to the true noise-free values.
- If we increase the number of matched points, will increase as well. Intuitively this is not right.

# Error in both images

- Residual error for noise in one image:

$$\epsilon_{\mathrm{res}} = \sqrt{\frac{1}{2n}\sum_{i=1}^{n} d(\mathbf{x_i}', \hat{\mathbf{x}}_\mathbf{i}')^2}$$

$$\hat{\mathbf{x}}_\mathbf{i}' = \hat{H}\bar{\mathbf{x}}_\mathbf{i}$$

- Residual error for noise in two images:

$$\epsilon_{\mathrm{res}} = \sqrt{\frac{1}{4n}\sum_{i=1}^{n} d(\mathbf{x_i}, \hat{\mathbf{x}}_\mathbf{i})^2 + \sum_{i=1}^{n} d(\mathbf{x_i}', \hat{\mathbf{x}}_\mathbf{i}')^2}$$

$$\hat{\mathbf{x}}_\mathbf{i}' = \hat{H}\hat{\mathbf{x}}_\mathbf{i}$$

# Optimal Estimators (MLE)

- Goal: derive formulae for the expected residual error of the Maximum Likelihood Estimate.
- Let $f$ be a function defining a mapping from $\mathbb{R}^M$ to $\mathbb{R}^N$ .
- Consider a point $\bar{\mathbf{X}} \in \mathbb{R}^N$ with a vector of parameters $\bar{\mathbf{P}} \in \mathbb{R}^M$ such that $f(\bar{\mathbf{P}}) = \bar{\mathbf{X}}$

# Optimal Estimators (MLE)

- In the context of 2D projectivities with measurements in the second image and a noise free set of points

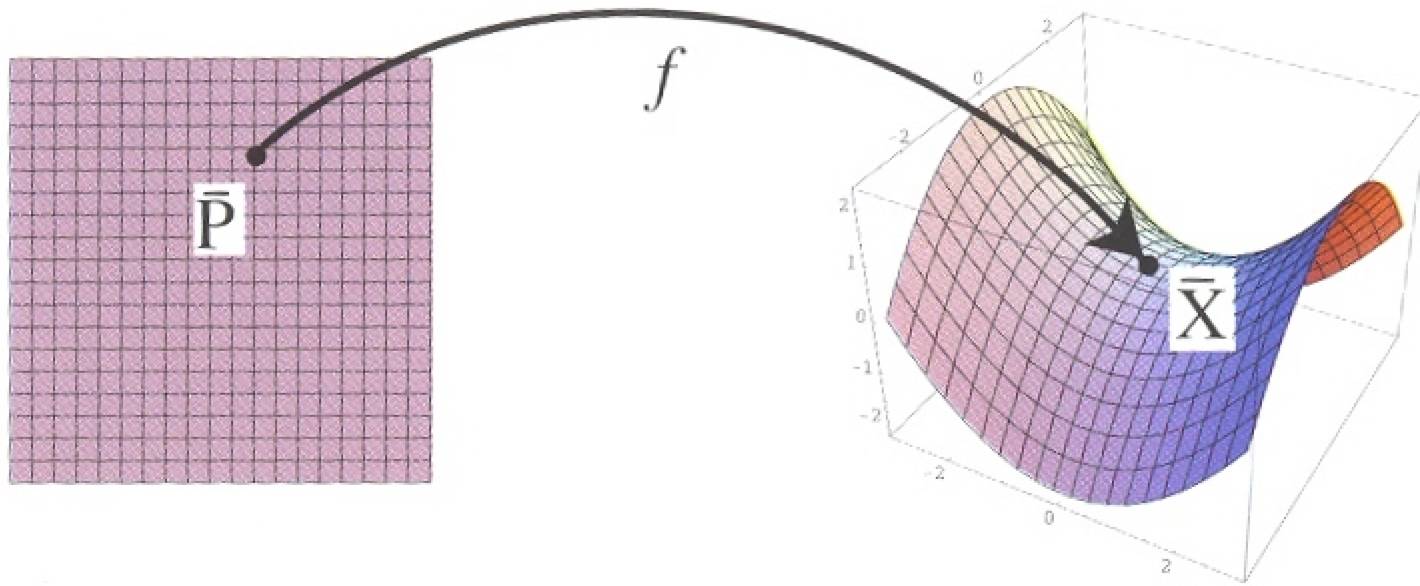
$$\bar{\mathbf{x_i}}' = \bar{H}\bar{\mathbf{x_i}}$$

we get:

$$f(\bar{\mathbf{P}}) = \bar{H}\bar{\mathbf{x_i}} \rightarrow \bar{\mathbf{X}} = \bar{\mathbf{x_i}}'$$

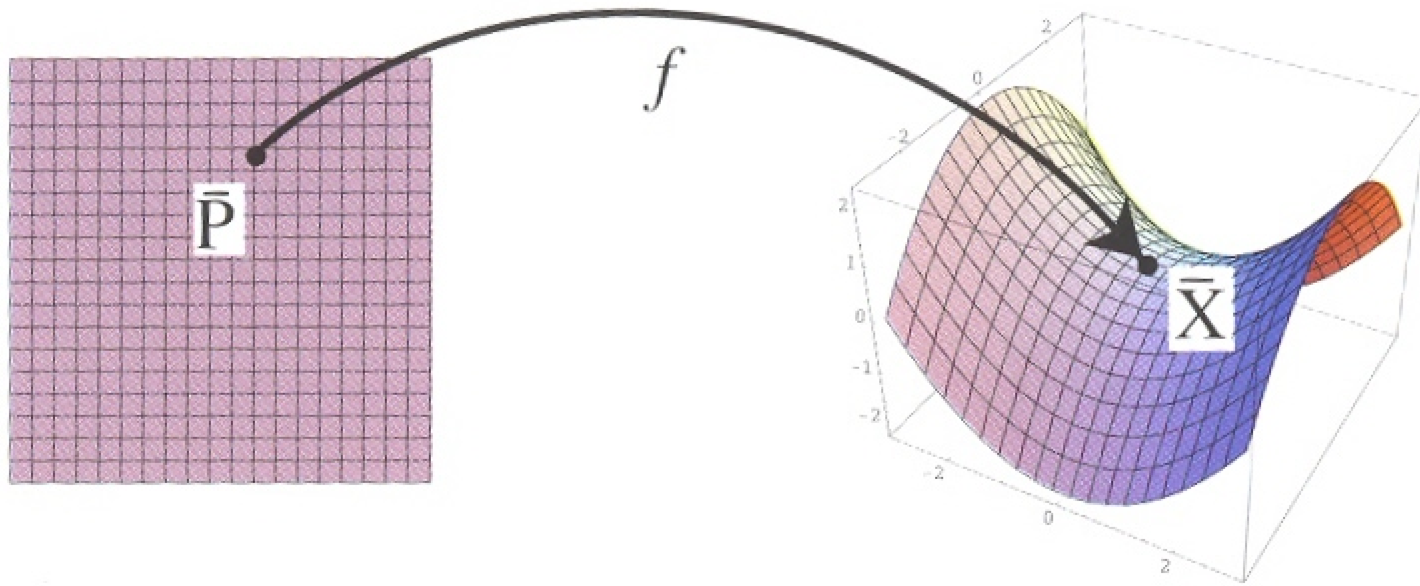
- Thus $f(\bar{\mathbf{P}})$ defines all possible projections from the given set of points from the first image $\bar{\mathbf{x_i}}$ to a set of points $\bar{\mathbf{X}}$ (representing $\bar{\mathbf{x_i}}'$) in the second image.

# Optimal Estimators (MLE)

- 

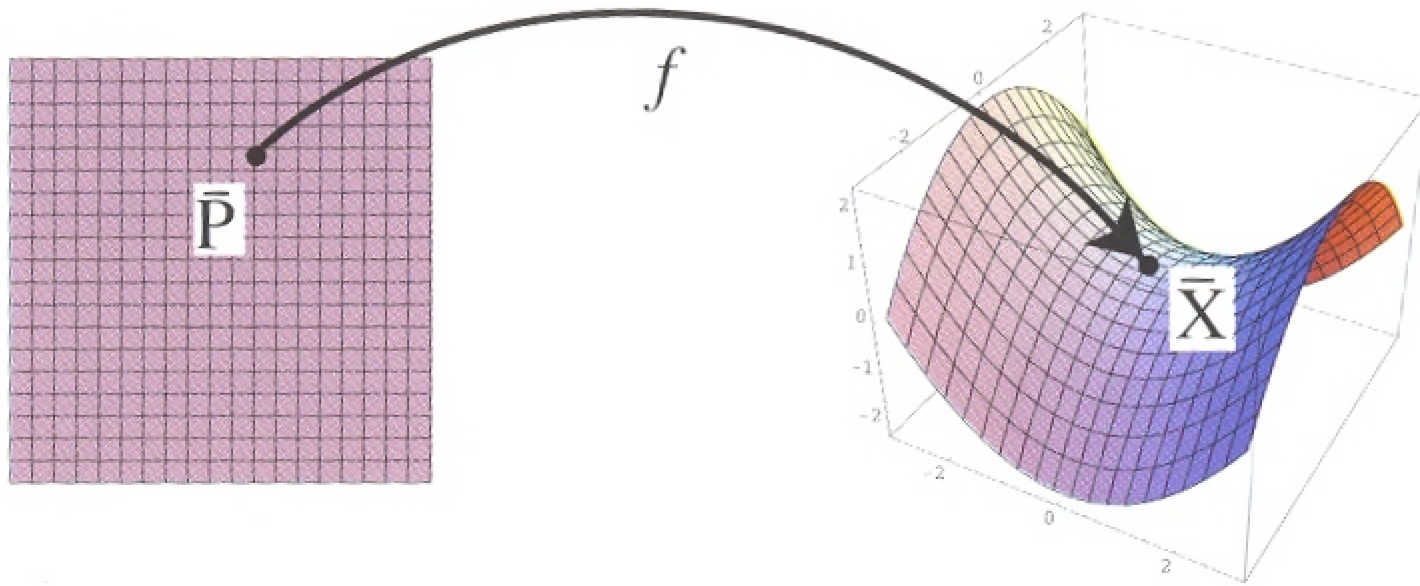

- Thus $f(\bar{\mathbf{P}})$ defines all possible projections from the given set of points from the first image $\bar{\mathbf{x}}_\mathbf{i}$ to a set of points $\bar{\mathrm{X}}$ (representing $\bar{\mathbf{x}}_\mathbf{i}'$) in the second image.

# Optimal Estimators (MLE)

- 


- The surface traced out by $f(\bar{\mathbf{P}})$ is called $S_M$.
- The dimension of this surface is $d$, where $d$ is the degrees of freedom. In this case it is 8 because of $H$ .

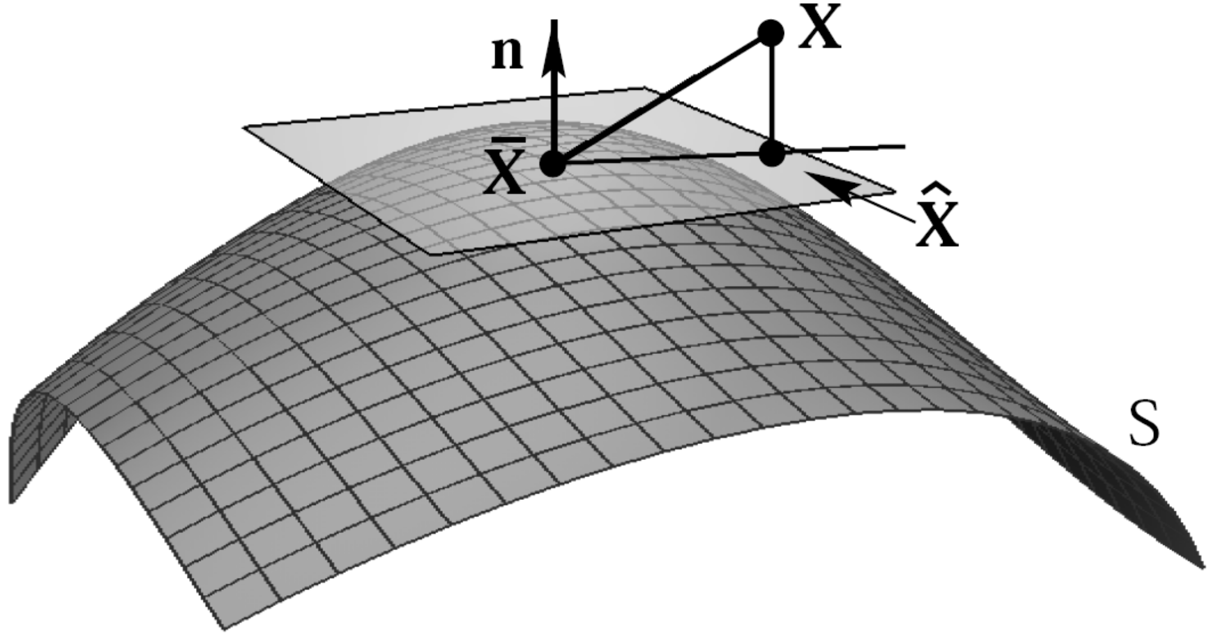
# Optimal Estimators (MLE)

- 


- Now given a measurement $\mathbf{X}$, the ML estimator returns the point on $S_M$ closest to $\mathbf{X}$
- Why on $S_M$ ? Well, because $S_M$ defines the only possible mappings (projections).

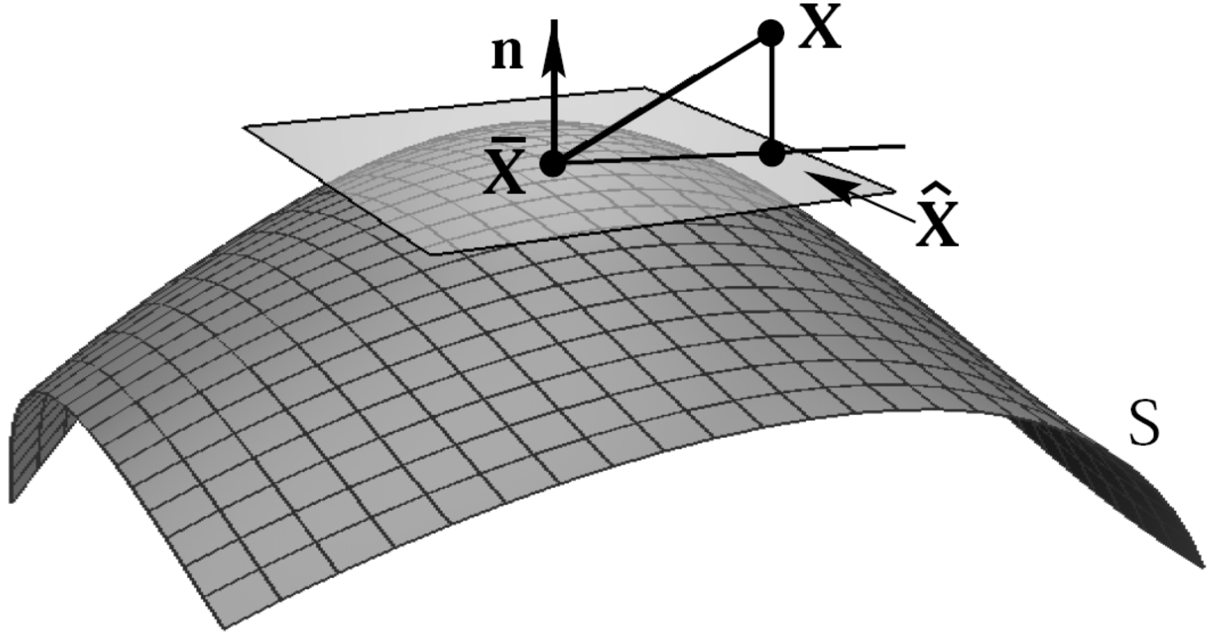
# Optimal Estimators (MLE)

- For our convenience we assume that in the neighbourhood of $\bar{X}$ the real surface can be approximated by the tangent surface.



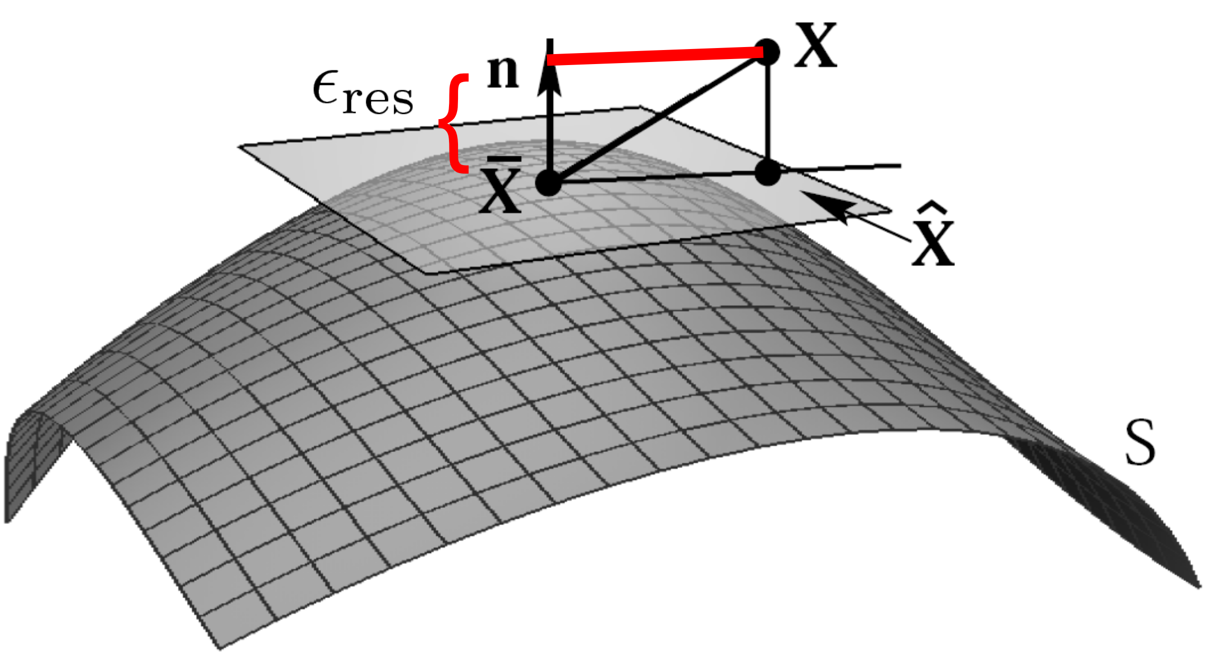
# Optimal Estimators (MLE)

- Now $\hat{\mathbf{X}}$ can be calculated by projecting $\mathbf{X}$ onto the tangent plane of $S_M$ .



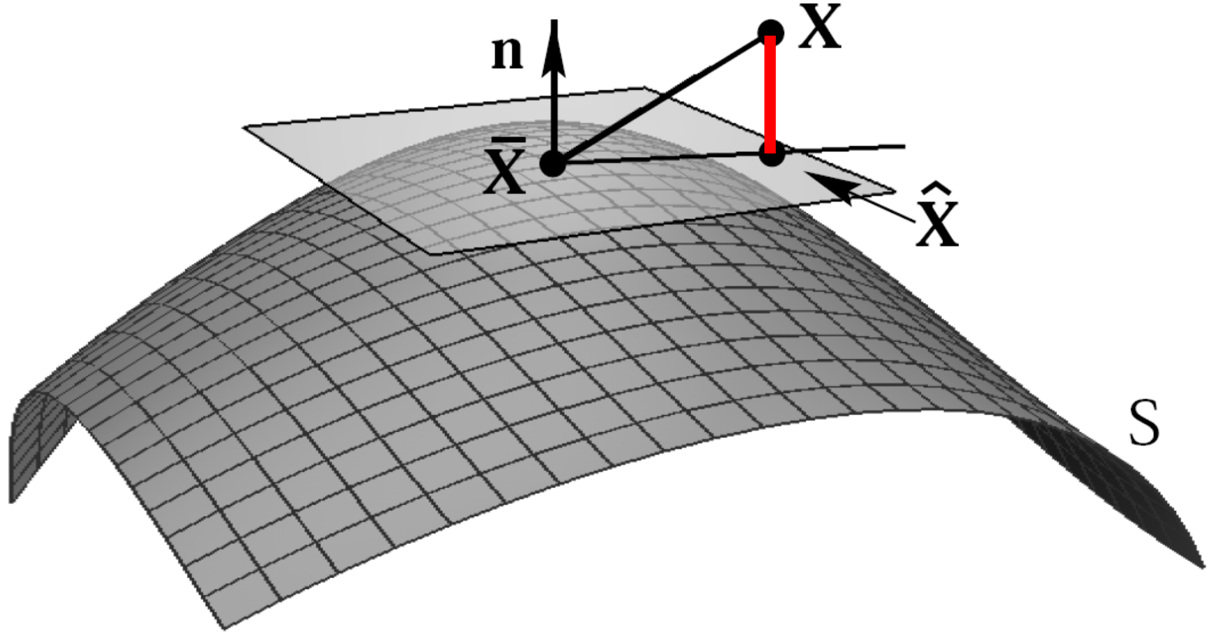
# Optimal Estimators (MLE)

- Geometrically: The residual error can be obtained by projecting $\mathbf{X}$ onto the normal of the tangent surface.



# Optimal Estimators (MLE)

- The *estimation error (* $d(\hat{\mathrm{x}}, \bar{\mathrm{x}})$ *)* can be obtained by projecting $\mathrm{X}$ onto the tangent surface of $S_M$.



# Optimal Estimators (MLE)

- Calculating the errors for a measurement assumed to have an isotropic distribution of the error.
- Result 5.1: The projection of an isotropic Gaussian distribution defined on $\mathbb{R}^N$ with total variance $N\sigma^2$ onto a subspace of dimension $s$ is an isotropic Gaussian distribution with total variance $s\sigma^2$.

# Optimal Estimators (MLE)

- Calculating the errors for a measurement assumed to have an isotropic distribution of the error.
- Result 5.1: The projection of an isotropic Gaussian distribution defined on $\mathbb{R}^N$ with total variance $N\sigma^2$ onto a subspace of dimension $s$ is an isotropic Gaussian distribution with total variance $s\sigma^2$.
  - Or geometrically: Any orthogonal projection of a hypersphere remains a hypersphere.

# Optimal Estimators (MLE)

- Result 5.2:
  - Residual error: $s = N - d$

$$\epsilon_{\mathrm{res}} = \sigma\sqrt{1 - \frac{d}{N}} = \sqrt{E[||\hat{\mathbf{X}} - \mathbf{X}||^2/N]}$$

  - Estimation error: $s = d$

$$\epsilon_{\mathrm{res}} = \sigma\sqrt{\frac{d}{N}} = \sqrt{E[||\hat{\mathbf{X}} - \bar{\mathbf{X}}||^2/N]}$$

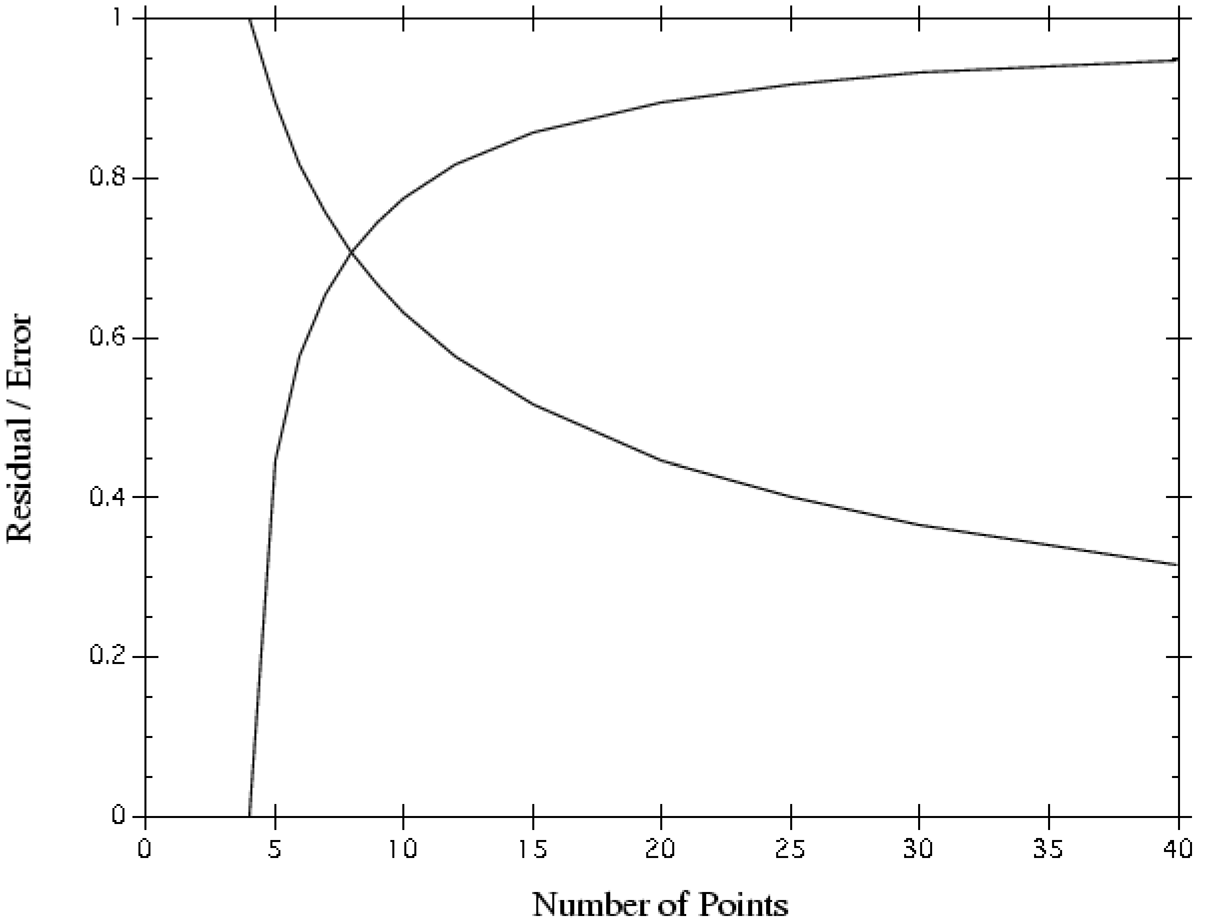
- These errors give lower bounds for error measures.

# Optimal Estimators (MLE)

- Example: Error in one image. d = 8, N = 2n:

$$\epsilon_{\text{res}} = \sigma\sqrt{1 - \frac{4}{n}}$$

$$\epsilon_{\text{est}} = \sigma\sqrt{\frac{4}{n}}$$

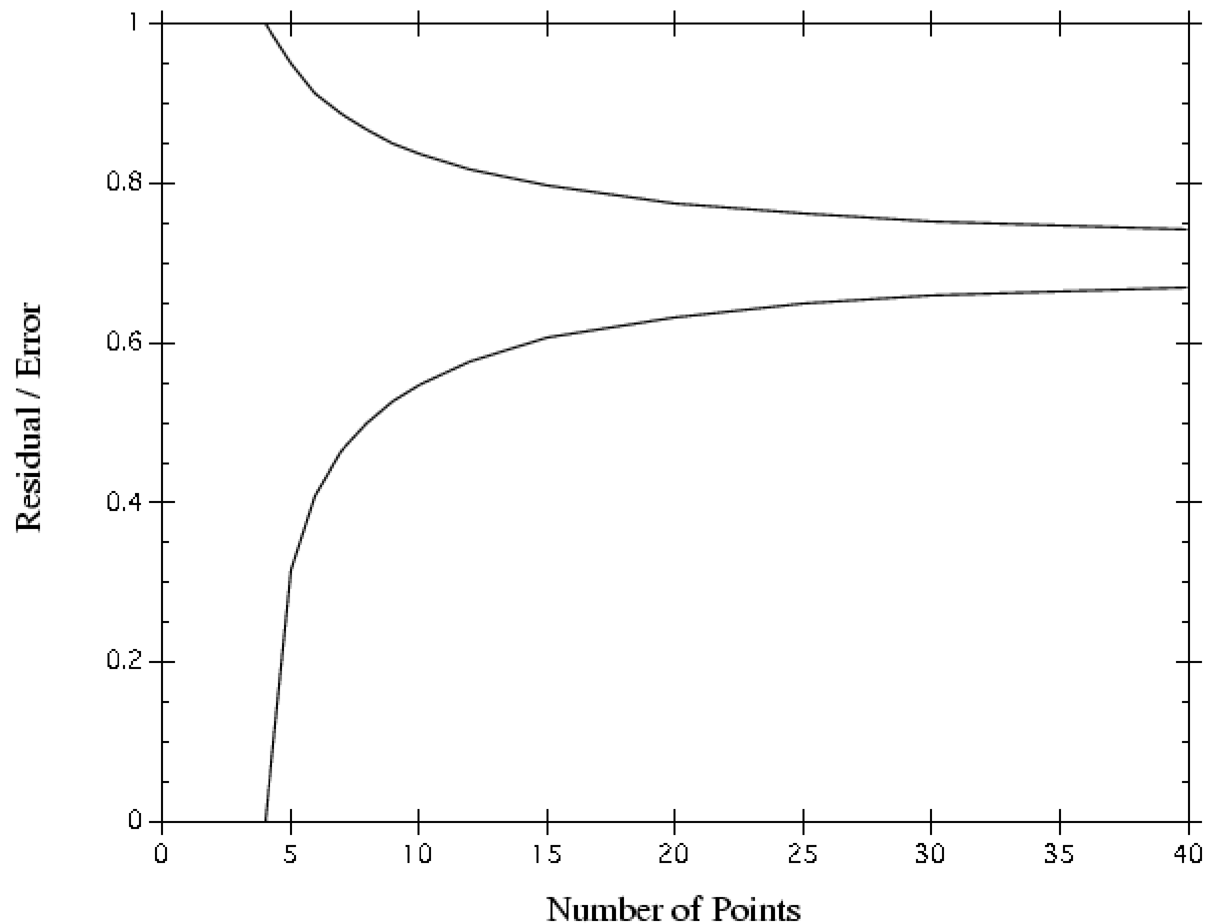


# Optimal Estimators (MLE)

- Example: Error in one image. d = 2n+ 8, N = 4n:

$$\epsilon_{\mathrm{res}} = \sigma\sqrt{\frac{n-4}{2n}}$$

$$\epsilon_{\mathrm{est}} = \sigma\sqrt{\frac{n+4}{2n}}$$



# Optimal Estimators (MLE)

- Mahalanobis distance
  - Residual error:

$$\epsilon_{\text{res}} = \sqrt{E[||\hat{\mathbf{X}} - \mathbf{X}||^2_\Sigma/N]}$$

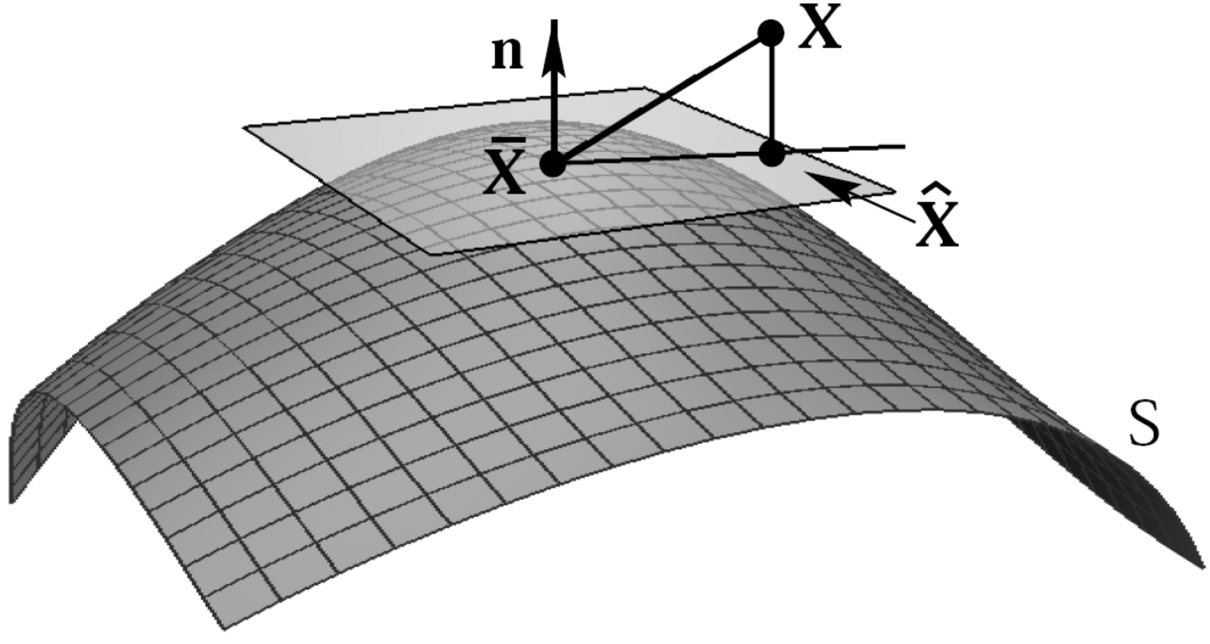
  - Estimation error:

$$\epsilon_{\text{res}} = \sqrt{E[||\hat{\mathbf{X}} - \bar{\mathbf{X}}||^2_\Sigma/N]}$$

# Optimal Estimators (MLE)

- Determining convergence of an algorithm

$$||\mathbf{X}-\bar{\mathbf{X}}||^2 = ||\mathbf{X}-\hat{\mathbf{X}}||^2 + ||\bar{\mathbf{X}}-\hat{\mathbf{X}}||^2$$



# Optimal Estimators (MLE)

- Determining convergence of an algorithm

$$||\mathbf{X}-\bar{\mathbf{X}}||^2 = ||\mathbf{X}-\hat{\mathbf{X}}||^2 + ||\bar{\mathbf{X}}-\hat{\mathbf{X}}||^2$$

- Unnecessary to determine degrees of freedom
- With repeated runs allows for success rate
- Can only be used when is known
- Relies on assumption that is locally planar. If equality not satisfied: Solution is not good or (less likely) planar assumption is wrong
- Test addresses *global* solution

# Covariance of the estimated transformation

- We can now calculate how well the algorithm behaved
- But we are still uncertain of the accuracy of the actual transformation:
  - If the selected points are close to a line, the transformation itself is not accurate, even though the residual and estimation error were minimized.
- Thus next we will calculate the uncertainty of the transformation, captured by a covariance matrix.

# Forward propagation of covariance

- To understand how we can calculate the covariance (or uncertainty) of the transformation, we will first investigate how the uncertainty of the transformation propagates onto the projected points.
- Then, with the backward propagation of covariance we will calculate the covariance (or uncertainty) of the transformation given image measurements.

# Forward propagation of covariance

- Result 5.3: affine transformation
  - Let $\mathbf{v}$ be a random vector in $\mathbb{R}^M$ with mean $\bar{\mathbf{v}}$ and covariance matrix $\Sigma$ , and suppose that $f : \mathbb{R}^M \rightarrow \mathbb{R}^N$ is an affine mapping defined by $f(\mathbf{v}) = f(\bar{\mathbf{v}}) + A(\mathbf{v} - \bar{\mathbf{v}})$ Then $f(\mathbf{v})$ is a random variable with mean $f(\bar{\mathbf{v}})$ and covariance matrix $A\Sigma A^T$

# Forward propagation of covariance

- Non-linear propagation
  - Let $\mathbf{v}$ be a random vector in $\mathbb{R}^M$ with mean $\bar{\mathbf{v}}$ and covariance matrix $\Sigma$, and let $f : \mathbb{R}^M \rightarrow \mathbb{R}^N$ be differentiable in a neighbourhood of $\bar{\mathbf{v}}$. Then up to a first order approximation, $f(\mathbf{v})$ is a random variable with mean $f(\bar{\mathbf{v}})$ and covariance $J\Sigma J^T$, where $J$ is the Jacobian matrix of $f$, evaluated at $\bar{\mathbf{v}}$.

# Forward propagation of covariance

- Example 5.4:
  - Mean is at the origin. $\Sigma$ = diag(1,4)

$$x' = f(x, y) = 3x + 2y - 7$$

# Forward propagation of covariance

- Example 5.4:
  - Mean is at the origin. $\Sigma$ = diag(1,4)

$$x' = f(x, y) = 3x + 2y - 7$$

  - A = [3 2]
  - Thus: variance of x' is $A \Sigma A^T = 25$

# Forward propagation of covariance

- Example 5.4: non-linear
  - Mean is at the origin. $\Sigma$ = diag(1,4)

$$x' = f(x, y) = 5x^2 + 3y^2 + 2xy + 3x + 2y - 7$$

# Forward propagation of covariance

- Example 5.4: non-linear
  - Mean is at the origin. $\Sigma$ = diag(1,4)

$$x' = f(x, y) = 5x^2 + 3y^2 + 2xy + 3x + 2y - 7$$

  - J at the origin = [3 2]
  - Thus: variance of x' is $J \Sigma J^T = 25$

# Forward propagation of covariance

- Example 5.7
  - Analytically:

$$\bar{x}' = 5 + \sigma^2$$

$$\sigma^2_{x'} = 25\sigma^2 + 2\sigma^4$$

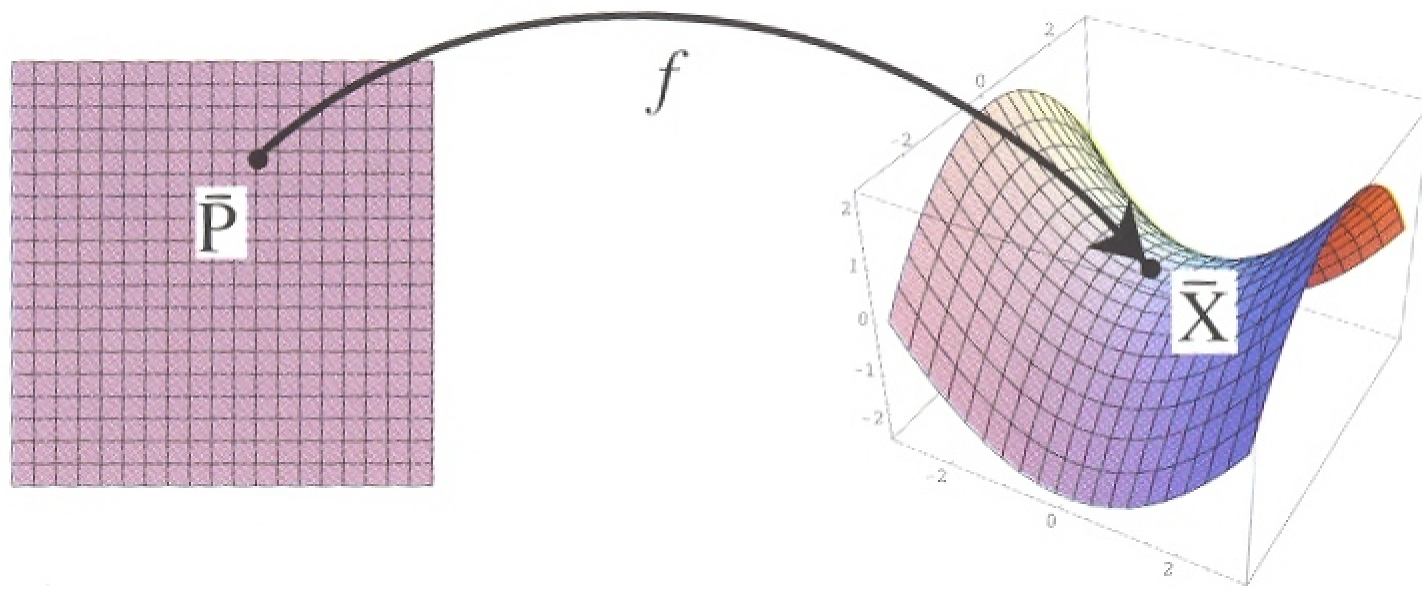
  - Approximation:

$$\bar{x}' = 5$$

$$\sigma^2_{x'} = 25\sigma^2$$

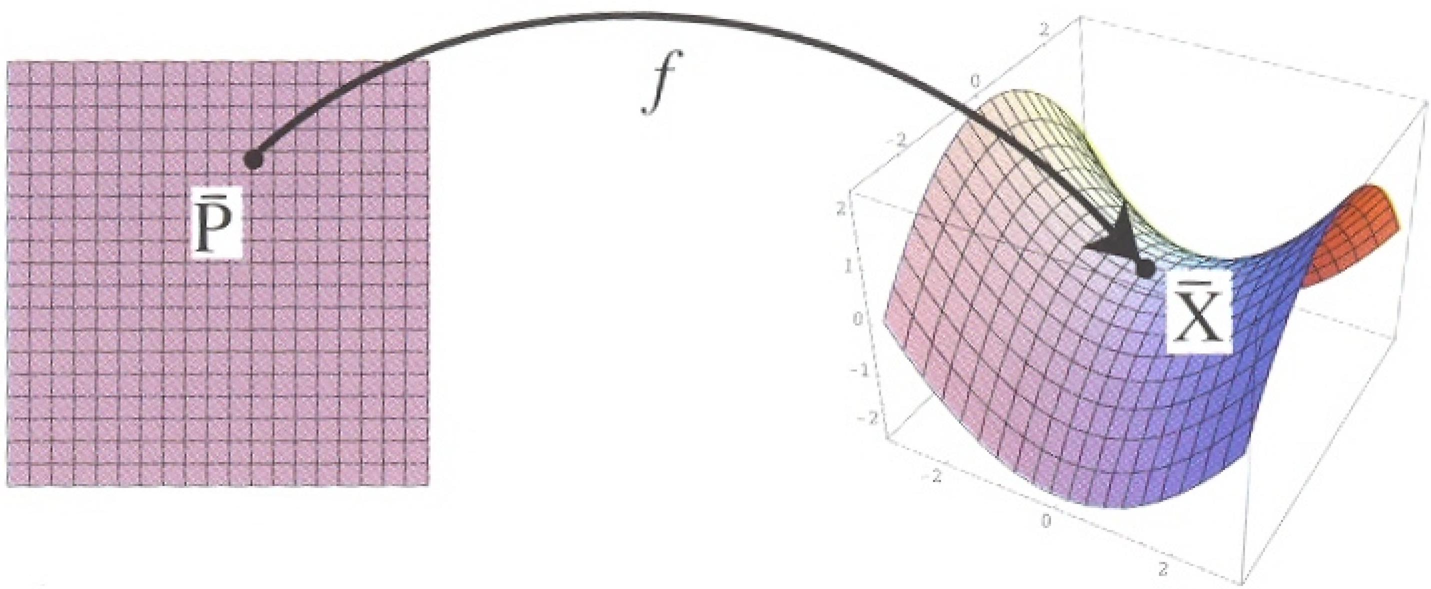
# Backward propagation of covariance

- Remember $f : \mathbb{R}^M \rightarrow S_M$



# Backward propagation of covariance

- Remember $f : \mathbb{R}^M \rightarrow S_M$

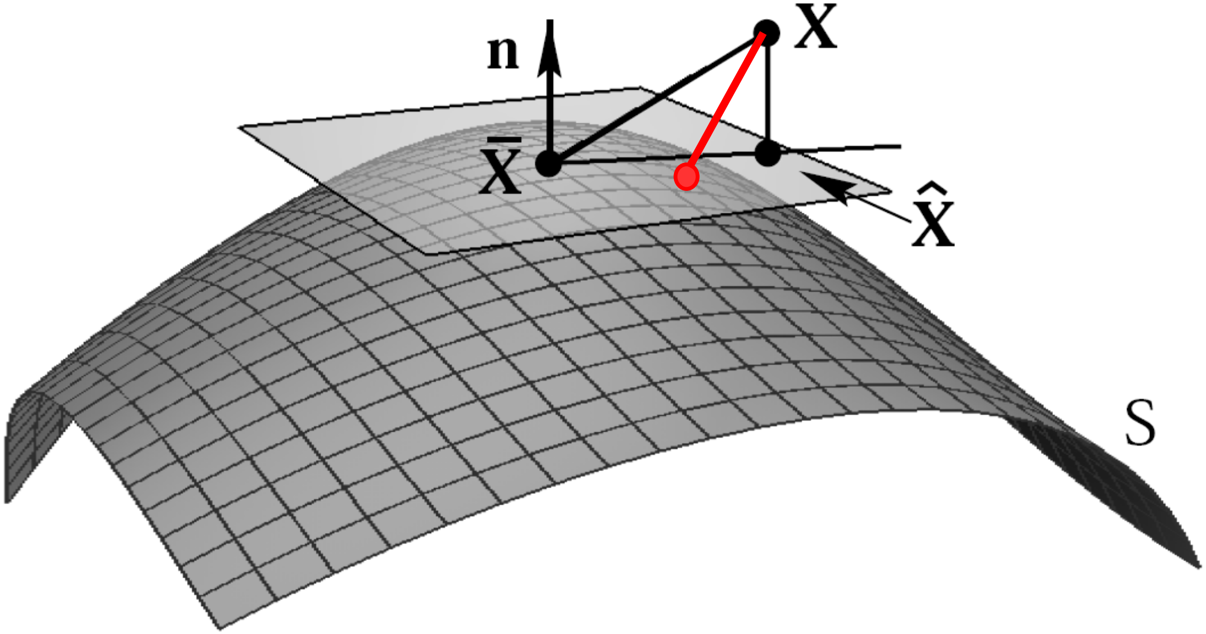



- $f$ is by assumption invertible on $S_M$, defined as:

$$f^{-1} : S_M \rightarrow \mathbb{R}^M$$

# Backward propagation of covariance

- We now define a mapping $\eta : \mathbb{R}^N \rightarrow S_M$ mapping $\mathbf{X}$ to its closest point on $S_M$

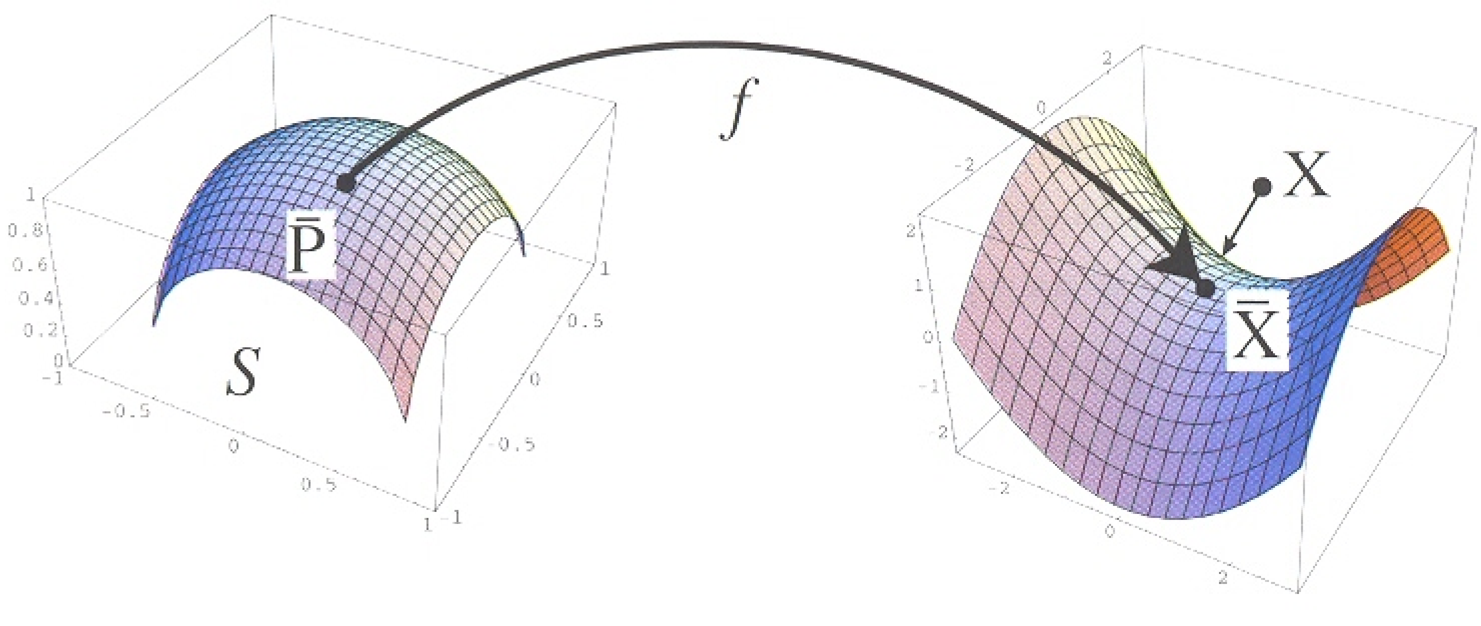


# Backward propagation of covariance

- We compose a composite map

$$f^{-1} \circ \eta : \mathbb{R}^N \rightarrow \mathbb{R}^M$$

which in the end we will use to propagate the covariance from $\mathbf{X}'$ to $\hat{\mathbf{X}}'$ on $S_M$ and finally to $\hat{\mathbf{h}}$



# Backward propagation of covariance

- Consider first that $f$ is an affine mapping.
- Since $f$ is affine, we can write

$$f(\mathbf{P}) = f(\bar{\mathbf{P}}) + J(\mathbf{P} - \bar{\mathbf{P}})$$

# Backward propagation of covariance

- Consider first that $f$ is an affine mapping.
- Since $f$ is affine, we can write

$$f(\bar{\mathbf{P}}) = \bar{\mathbf{X}} \qquad f(\mathbf{P}) = f(\bar{\mathbf{P}}) + J(\mathbf{P} - \bar{\mathbf{P}})$$

# Backward propagation of covariance

- Consider first that $f$ is an affine mapping.
- Since $f$ is affine, we can write

$$f(\bar{\mathbf{P}}) = \bar{\mathbf{X}} \qquad f(\mathbf{P}) = f(\bar{\mathbf{P}}) + J(\mathbf{P} - \bar{\mathbf{P}})$$

- ML tries to minimize:

$$||\mathbf{X} - \hat{\mathbf{X}}||_\Sigma = ||\mathbf{X} - f(\hat{\mathbf{P}})||_\Sigma$$

# Backward propagation of covariance

- Consider first that $f$ is an affine mapping.
- Since $f$ is affine, we can write

$$f(\bar{\mathbf{P}}) = \bar{\mathbf{X}} \qquad f(\mathbf{P}) = f(\bar{\mathbf{P}}) + J(\mathbf{P} - \bar{\mathbf{P}})$$

- ML tries to minimize:

$$||\mathbf{X} - \hat{\mathbf{X}}||_\Sigma = ||\mathbf{X} - f(\hat{\mathbf{P}})||_\Sigma$$

# Backward propagation of covariance

- Consider first that $f$ is an affine mapping.
- Since $f$ is affine, we can write

$$f(\bar{\mathbf{P}}) = \bar{\mathbf{X}} \qquad f(\mathbf{P}) = f(\bar{\mathbf{P}}) + J(\mathbf{P} - \bar{\mathbf{P}})$$

- ML tries to minimize:

$$||\mathbf{X} - \hat{\mathbf{X}}||_\Sigma = ||\mathbf{X} - f(\hat{\mathbf{P}})||_\Sigma$$

$$||(\mathbf{X} - \bar{\mathbf{X}}) - J(\hat{\mathbf{P}} - \bar{\mathbf{P}})||_\Sigma$$

# Backward propagation of covariance

- Consider first that $f$ is an affine mapping.
- Since $f$ is affine, we can write

$$f(\bar{\mathbf{P}}) = \bar{\mathbf{X}} \qquad f(\mathbf{P}) = f(\bar{\mathbf{P}}) + J(\mathbf{P} - \bar{\mathbf{P}})$$

- ML tries to minimize:

$$||\mathbf{X} - \hat{\mathbf{X}}||_\Sigma = ||\mathbf{X} - f(\hat{\mathbf{P}})||_\Sigma$$

$$||(\mathbf{X} - \bar{\mathbf{X}}) - J(\hat{\mathbf{P}} - \bar{\mathbf{P}})||_\Sigma$$

$$(\hat{\mathbf{P}} - \bar{\mathbf{P}}) = (J^T\Sigma^{-1}J)^{-1}J^T\Sigma^{-1}(\mathbf{X} - \bar{\mathbf{X}})$$

# Backward propagation of covariance

- Consider first that $f$ is an affine mapping.
- Since $f$ is affine, we can write

$$f(\bar{\mathbf{P}}) = \bar{\mathbf{X}} \qquad f(\mathbf{P}) = f(\bar{\mathbf{P}}) + J(\mathbf{P} - \bar{\mathbf{P}})$$

- ML tries to minimize:

$$||\mathbf{X} - \hat{\mathbf{X}}||_\Sigma = ||\mathbf{X} - f(\hat{\mathbf{P}})||_\Sigma$$

Pseudo inverse w.r.t. $\Sigma$ cf:

$$(A^T A)^{-1} A^T$$

$$||(\mathbf{X} - \bar{\mathbf{X}}) - J(\hat{\mathbf{P}} - \bar{\mathbf{P}})||_\Sigma$$

$$(\hat{\mathbf{P}} - \bar{\mathbf{P}}) = (J^T \Sigma^{-1} J)^{-1} J^T \Sigma^{-1} (\mathbf{X} - \bar{\mathbf{X}})$$

# Backward propagation of covariance

- $$(\hat{\mathbf{P}} - \bar{\mathbf{P}}) = (J^T \Sigma^{-1} J)^{-1} J^T \Sigma^{-1} (\mathbf{X} - \bar{\mathbf{X}})$$
- If we write $\bar{\mathbf{P}} = f^{-1} \bar{\mathbf{X}}$ and $\hat{\mathbf{P}} = f^{-1} \bar{\mathbf{X}}$

$$f^{-1} \circ \eta(\mathbf{X}) = \hat{\mathbf{P}}$$

# Backward propagation of covariance

- $$(\hat{\mathbf{P}} - \bar{\mathbf{P}}) = (J^T \Sigma^{-1} J)^{-1} J^T \Sigma^{-1} (\mathbf{X} - \bar{\mathbf{X}})$$
- If we write $\bar{\mathbf{P}} = f^{-1} \bar{\mathbf{X}}$ and $\hat{\mathbf{P}} = f^{-1} \bar{\mathbf{X}}$

$$f^{-1} \circ \eta(\mathbf{X}) = \hat{\mathbf{P}}$$

$$= (J^T \Sigma^{-1} J)^{-1} J^T \Sigma^{-1} (\mathbf{X} - \bar{\mathbf{X}}) + f^{-1}(\bar{\mathbf{X}})$$

# Backward propagation of covariance

- $$(\hat{\mathbf{P}} - \underline{\bar{\mathbf{P}}}) = (J^T \Sigma^{-1} J)^{-1} J^T \Sigma^{-1} (\mathbf{X} - \bar{\mathbf{X}})$$
- If we write $\underline{\bar{\mathbf{P}} = f^{-1} \bar{\mathbf{X}}}$ and $\hat{\mathbf{P}} = f^{-1} \bar{\mathbf{X}}$

$$f^{-1} \circ \eta(\mathbf{X}) = \hat{\mathbf{P}}$$

$$= (J^T \Sigma^{-1} J)^{-1} J^T \Sigma^{-1} (\mathbf{X} - \bar{\mathbf{X}}) + \underline{f^{-1}(\bar{\mathbf{X}})}$$

$$= (J^T \Sigma^{-1} J)^{-1} J^T \Sigma^{-1} (\mathbf{X} - \bar{\mathbf{X}}) + f^{-1} \circ \eta(\bar{\mathbf{X}})$$

# Backward propagation of covariance

- $$(\hat{\mathbf{P}} - \bar{\mathbf{P}}) = (J^T \Sigma^{-1} J)^{-1} J^T \Sigma^{-1} (\mathbf{X} - \bar{\mathbf{X}})$$
- If we write $\bar{\mathbf{P}} = f^{-1} \bar{\mathbf{X}}$ and $\hat{\mathbf{P}} = f^{-1} \bar{\mathbf{X}}$

$$f^{-1} \circ \eta(\mathbf{X}) = \hat{\mathbf{P}}$$

$$= (J^T \Sigma^{-1} J)^{-1} J^T \Sigma^{-1} (\mathbf{X} - \bar{\mathbf{X}}) + f^{-1}(\bar{\mathbf{X}})$$

$$= (J^T \Sigma^{-1} J)^{-1} J^T \Sigma^{-1} (\mathbf{X} - \bar{\mathbf{X}}) + f^{-1} \circ \eta(\bar{\mathbf{X}})$$

- This proves that $f^{-1} \circ \eta$ is affine.

# Backward propagation of covariance

- Result 5.9:

  For an affine mapping of the form

  $$f(\mathbf{P}) = f(\bar{\mathbf{P}}) + J(\mathbf{P} - \bar{\mathbf{P}})$$

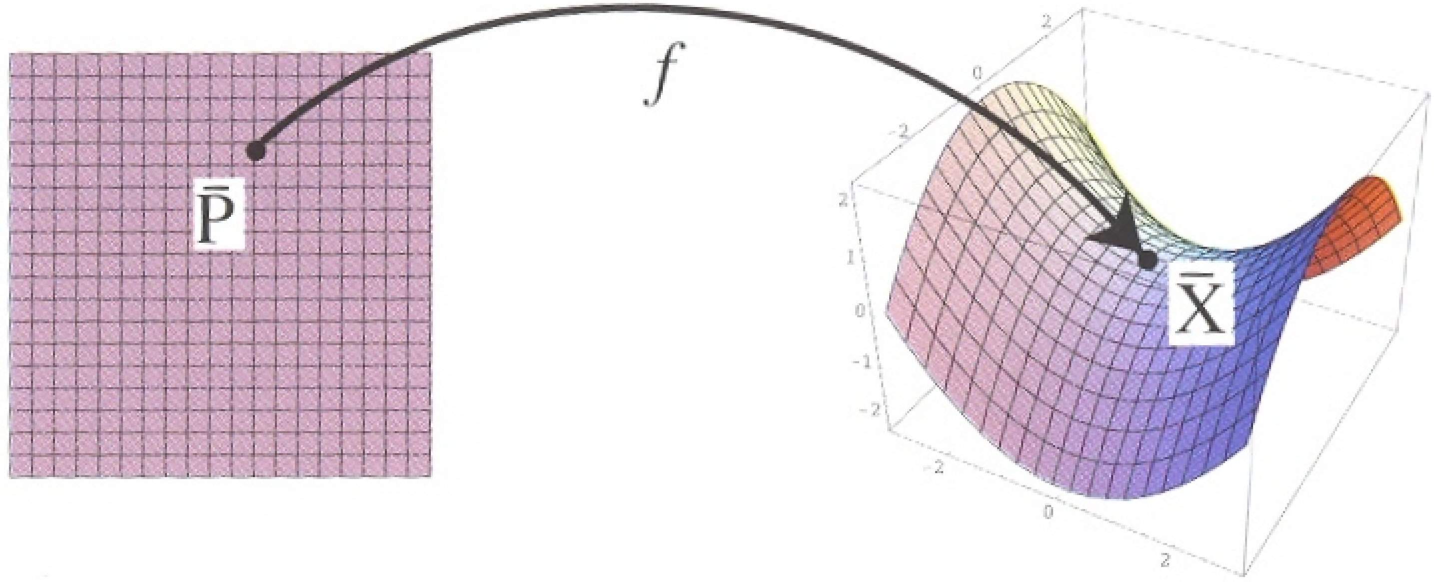
  the covariance matrix is given by

  $$\Sigma_{\mathbf{P}} = (J^T \Sigma_{\mathbf{X}}^{-1} J)^{-1}$$

# Backward propagation of covariance

- Over-parametrization:
  - The parameter space has d degrees of freedom, d < M. E.g. with projection matrix h: 8 < 9.

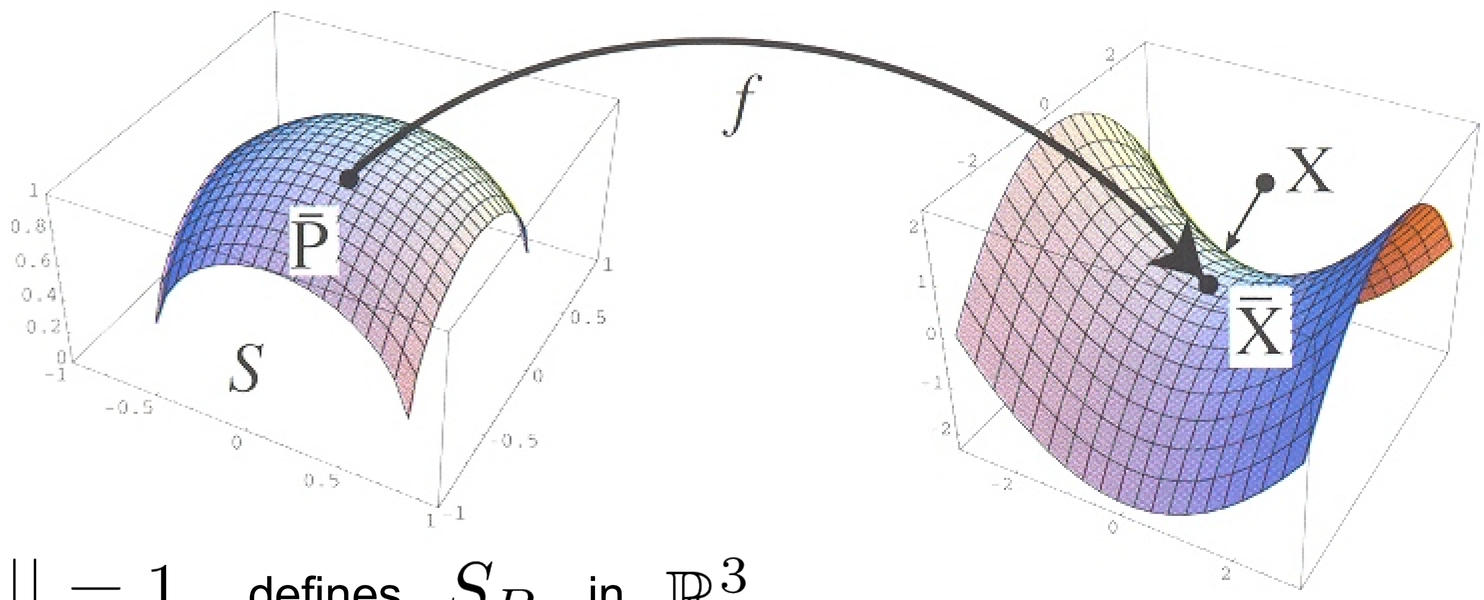
$$f(\mathbf{P}) = f(k\mathbf{P})$$



# Backward propagation of covariance

- Over-parametrization:
  - The parameter space has d degrees of freedom, d < M. E.g. with projection matrix h: 8 < 9.

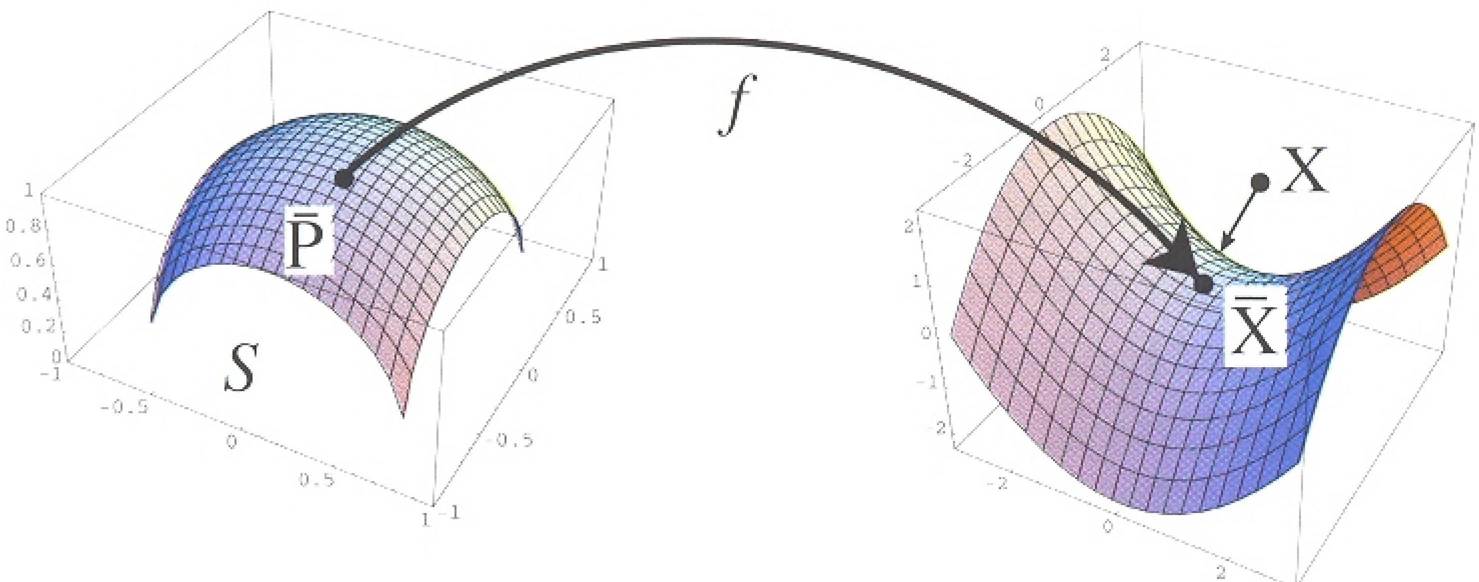
$$f(\mathbf{P}) = f(k\mathbf{P})$$




e.g. $||\mathbf{P}|| = 1$ defines $S_P$ in $\mathbb{R}^3$

# Backward propagation of covariance

- Over-parametrization: Result 5.11
  - define g: $g : \mathbb{R}^d \rightarrow \mathbb{R}^M$
  - define $f \circ g : \mathbb{R}^d \rightarrow \mathbb{R}^N$



# Backward propagation of covariance

- Over-parametrization: Result 5.11
  - define g: $g : \mathbb{R}^d \rightarrow \mathbb{R}^M$
  - define $f \circ g : \mathbb{R}^d \rightarrow \mathbb{R}^N$
  - partial derivative $f$ : $J$
  - partial derivative $g$ : $A$
  - partial derivative $f \circ g$ : $JA$

# Backward propagation of covariance

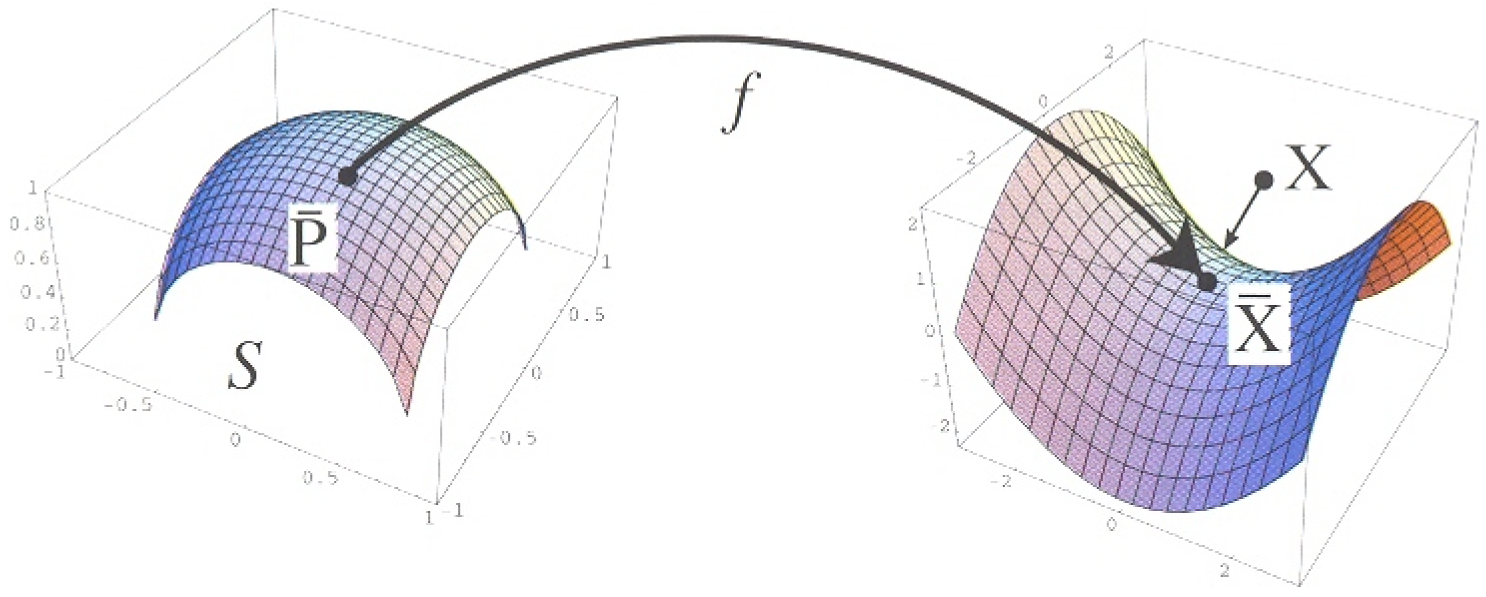
- Over-parametrization: Result 5.11
  - define g: $g : \mathbb{R}^d \rightarrow \mathbb{R}^M$
  - define $f \circ g : \mathbb{R}^d \rightarrow \mathbb{R}^N$
  - partial derivative $f$ : $J$
  - partial derivative $g$ : $A$
  - partial derivative $f \circ g$ : $JA$
  - result 5.10: $\Sigma_{\mathbf{P}} = (J^T \Sigma_{\mathbf{X}}^{-1} J)^{-1}$
  - for this results in: $\Sigma^{\mathbb{R}^N \rightarrow \mathbb{R}^d} = (A^T J^T \Sigma^{-1} J A)^{-1}$

# Backward propagation of covariance

- Over-parametrization: Result 5.11
  - define g: $g : \mathbb{R}^d \rightarrow \mathbb{R}^M$
  - define $f \circ g : \mathbb{R}^d \rightarrow \mathbb{R}^N$
  - partial derivative $f$ : $J$
  - partial derivative $g$ : $A$
  - partial derivative $f \circ g$ : $JA$
  - result 5.10: $\Sigma_{\mathbf{P}} = (J^T \Sigma_{\mathbf{X}}^{-1} J)^{-1}$
  - for $f \circ g$ this results in: $\Sigma^{\mathbb{R}^N \rightarrow \mathbb{R}^d} = (A^T J^T \Sigma^{-1} J A)^{-1}$
  - projection forwards $\mathbb{R}^d \rightarrow \mathbb{R}^M$ (using 5.6: $\Sigma' = J \Sigma J^T$)
  - $\Sigma^{\mathbb{R}^N \rightarrow \mathbb{R}^M} = A(A^T J^T \Sigma^{-1} J A)^{-1} A^T = (J^T \Sigma^{-1} J)^{\dagger A}$

# Backward propagation of covariance

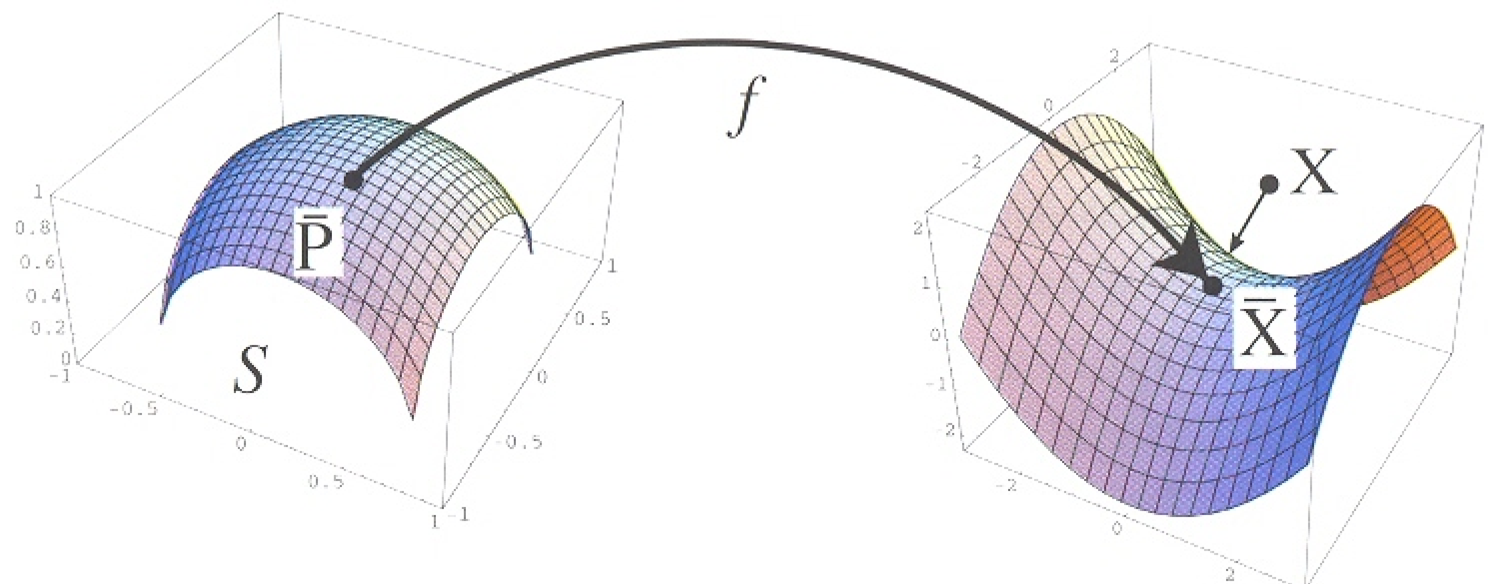
- Over-parametrization: Result 5.11




$$\Sigma^{\mathbb{R}^N \rightarrow \mathbb{R}^M} = A(A^T J^T \Sigma^{-1} J A)^{-1} A^T = (J^T \Sigma^{-1} J)^{\dagger A}$$

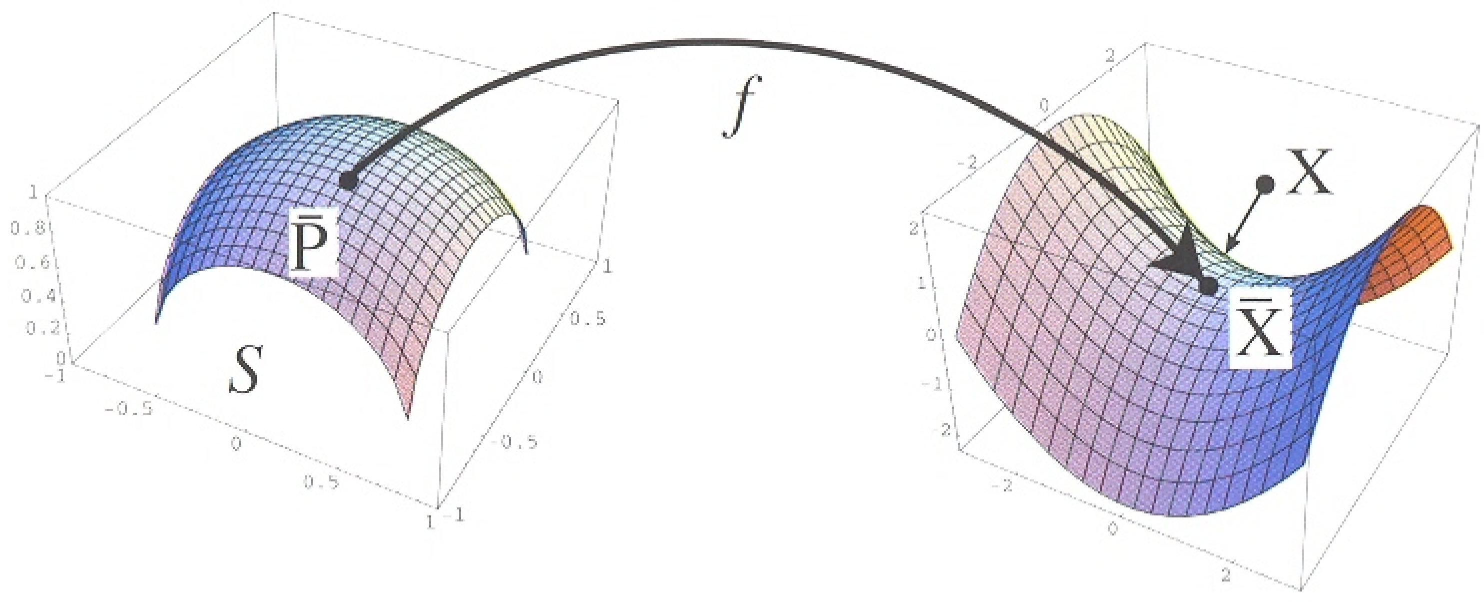
# Application and examples

- Error in one image.
- The function $f:\mathbb{R}^9 \rightarrow \mathbb{R}^{2n}$ maps h to the coordinates of the points $\mathbf{x_i}' = H\bar{\mathbf{x}}_\mathbf{i}$ .



# Application and examples

- Error in one image.
- The function $f : \mathbb{R}^9 \rightarrow \mathbb{R}^{2n}$ maps h to the coordinates of the points $\mathbf{x_i}' = H\bar{\mathbf{x}}_\mathbf{i}$ .
- $f(\mathbf{h})$ traces out $S_M$ (!) in $\mathbb{R}^N = \mathbb{R}^{2n}$



# Application and examples

- Error in one image.
  - Estimate transformation matrix $\hat{H}$
  - Compute Jacobian $J_f = \dfrac{\partial \mathbf{X}'}{\partial \mathbf{h}}$ at $\hat{\mathbf{h}}$

$$\mathrm{J}_i = \partial \mathbf{x}'_i / \partial \mathbf{h} = \frac{1}{w'_i} \begin{bmatrix} \tilde{\mathbf{x}}_i^\top & \mathbf{0}^\top & -x'_i \tilde{\mathbf{x}}_i^\top \\ \mathbf{0}^\top & \tilde{\mathbf{x}}_i^\top & -y'_i \tilde{\mathbf{x}}_i^\top \end{bmatrix}$$

  - Compute covariance of h by result 5.12:

$$\Sigma_{\mathbf{h}} = (J_f^T \Sigma_{\mathbf{X}'}^{-1} J_f)^{\dagger}$$

# Application and examples

- Error in one image.

$$\Sigma_h = (J^\top J)^{+A_1} = A_1\left(A_1^\top (J^\top J) A_1\right)^{-1} A_1^\top = \frac{1}{18}\left[\begin{array}{ccc|ccc|ccc} 5 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & -1 \\ 0 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 9 & 0 & 0 & 0 & 9 & 0 & 0 \\ \hline 0 & 0 & 0 & 9 & 0 & 0 & 0 & 0 & 0 \\ -4 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 9 & 0 & 9 & 0 \\ \hline 0 & 0 & 9 & 0 & 0 & 0 & 18 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9 & 0 & 18 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 2 \end{array}\right]. \tag{5.14}$$

# Application and examples

- Error in both images.
  - applying result 5.12:

$$J^{\top}\Sigma_{\mathbf{X}}^{-1}J = \begin{bmatrix} A^{\top}\Sigma_{\mathbf{X}}^{-1}A & A^{\top}\Sigma_{\mathbf{X}}^{-1}B \\ B^{\top}\Sigma_{\mathbf{X}}^{-1}A & B^{\top}\Sigma_{\mathbf{X}}^{-1}B \end{bmatrix}$$

  - In this case

$$\Sigma_{\mathbf{h}} = 2(J_f^T \Sigma_{\mathbf{X}'}^{-1} J_f)^{\dagger}$$

# Application and examples

- Using the covariance matrix to evaluate the uncertainty of a point transfer.

$$\Sigma_{\mathbf{x}'} = J_{\mathbf{h}} \Sigma_{\mathbf{h}} J_{\mathbf{h}}^T + J_{\mathbf{X}} \Sigma_{\mathbf{X}} J_{\mathbf{X}}^T$$

uncertainty of projection matrix

uncertainty of measurement

# Application and examples

- Example 5.14

$$\Sigma_{\mathbf{x}'} = \begin{bmatrix} \sigma_{x'x'} & \sigma_{x'y'} \\ \sigma_{x'y'} & \sigma_{y'y'} \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 2 - x^2 + x^4 + y^2 + x^2y^2 & xy(x^2 + y^2 - 2) \\ xy(x^2 + y^2 - 2) & 2 - y^2 + y^4 + x^2 + x^2y^2 \end{bmatrix}.$$

  - The fourth power of $\sigma_{x'x'}$ shows that extrapolating values of transformed points far outside the set of points used to calculate the transformation is not reliable.

# Monte Carlo Estimation

- Alternate estimation of H and $\Sigma_{\mathbf{H}}$ , and $\Sigma_{\mathbf{X}}$ and $\mathbf{X}$ .