Pontifícia Universidade Católica
do Rio de Janeiro

David Rey

Distributions and Immersions

MSc Thesis

Thesis presented to the Post–graduate Program in Applied Mathematics of the Mathematics Department, PUC–Rio as partial fulfillment of the requirements for the degree of Master in Applied Mathematics

Adviser: Prof. Thomas Lewiner

Rio de Janeiro
November 2007

Pontifícia Universidade Católica
do Rio de Janeiro

David Rey

Distributions and Immersions

Thesis presented to the Post–graduate Program in Applied Mathematics of the Mathematics Department, PUC–Rio as partial fulfillment of the requirements for the degree of Master in Applied Mathematics. Approved by the following commission:

Prof. Thomas Lewiner
Adviser
Department of Mathematics — PUC–Rio

Prof. Henri Anciaux
Department of Mathematics — PUC–Rio

Prof. Marcos Craizer
Department of Mathematics — PUC–Rio

Prof. José Eugenio Leal
Head of the Science and Engineering Center — PUC–Rio

Rio de Janeiro — November 29, 2007

David Rey

David Rey received his B.Sc from Université Montpellier II in Electrical Engineering, with minor in Opto-Electronics. He worked during a semester at the Gávea Sensor laboratory inside the Pontifícia Universidade Católica do Rio de Janeiro on fiber optic sensors, before entering the Mathematics department in 2006.

Bibliographic data

Rey, David

Distributions and Immersions / David Rey; adviser: Thomas Lewiner. — Rio de Janeiro : PUC–Rio, Department of Mathematics, 2007.

v., 60 f: il. ; 29,7 cm

1. MsC Thesis - Pontifícia Universidade Católica do Rio de Janeiro, Department of Mathematics.

Bibliography included.

1. Mathematics – Dissertation. 2. Differential Geometry. 3. Immersions. 4. Distribution Theory. 5. Geometric Singularities. 6. Discrete Differential Geometry. I. Lewiner, Thomas. II. Pontifícia Universidade Católica do Rio de Janeiro. Department of Mathematics. III. Title.

CDD: 510

Acknowledgments

Fisrt of all, I would like to thank the CAPES and the PUC-Rio, for the financial support, without which this work would not have been realized. I also would like to thank José Eugenio Leal for helping my master defense, allowing it to occur on time. My greatest acknowledgement goes to Thomas Lewiner, he was there on my first day at the mathematic departement of PUC-Rio and guided me in my choices. He ended up offering me my master disseration subject and there are no words to describe his behaviour as an adviser, a teacher and a friend to me althrough my master. Merci Thomas! In a similar way, I also want to thank Henri Anciaux for his support and his patience, and for contributing many times in this work. As being a member of my jury, I would like to thank Marcos Craizer. His observations have been of big interest to me and his support along my work was really constructive. Thank you Creuza, Katia, Otávio and Orlando you were all very nice to me, and helped me many times through my stay at the mathematic departement. I want to give a special thank to Renato for the extra-academic hand he gave me, valeu pela casa! Thank you to all the colleagues, particularly Eduardo, Zé, Maria-Clara, Joana, Camilla, Betina, Alex, Fabiano, Rener, Thalès, Yuri and I certainly miss some of them. I am very grateful to Carla Kato, Alexandre Ribeiro and Sully from the fiber optic sensor laboratory for their support along my time in Rio. I also want to name my friends Tiago, Breno, Sol, Greg, Elisa, and especially Irene for being there all the time. I will not forget to thank my parents, who are the most supporting parents a son can wish to have, thank you for giving me the opportunity to do such a thing. Thank you Brother, thank you Sister, and thank you to all my familly and my friends that live abroad, Laurent, Arnaud, Violaine, la Blonde and the Greeks.

Abstract

Rey, David; Lewiner, Thomas. **Distributions and Immersions**. Rio de Janeiro, 2007. 60p. MScThesis — Department of Mathematics, Pontifícia Universidade Católica do Rio de Janeiro.

The challenge of studying shapes has led mathematicians to create powerful abstract concepts, in particular through Differential Geometry. However, differential tools do not apply to simple shapes like cubes. This work is an attempt to use modern advances of the Analysis, namely Distribution Theory, to extend differential quantities to singular objects. Distributions generalize functions, while allowing infinite differentiation. The substitution of classical immersions, which usually serve as submanifold parameterizations, by distributions might thus naturally generalize smooth immersion. This leads to the concept of $\mathcal{D}$-immersion. This work proves that this formulation actually generalizes smooth immersions. Extensions to non-smooth of immersions are discussed through examples and specific cases.

Keywords

Differential Geometry. Immersions. Distribution Theory. Geometric Singularities. Discrete Differential Geometry.

Resumo

Rey, David; Lewiner, Thomas. **Distribuições e Imersões**. Rio de Janeiro, 2007. 60p. Dissertação de Mestrado — Departamento de Matemática, Pontifícia Universidade Católica do Rio de Janeiro.

Os desafios de estudar formas levaram matemáticos a criar abstrações, em particular através da geometria diferencial. Porém, formas simples como cubos não se adequam a ferramentas diferenciáveis. Este trabalho é uma tentativa de usar avanços recentes da análise, no caso a teoria das distribuições, para estender quantidades diferenciáveis a objetos singulares. Como as distribuições generalizam as funções e permitem derivações infinitas, a substituição das parametrizações de subvariedades clássicas por distribuições poderia naturalmente generalizar as subvariedades suaves. Isso nos leva a definir $\mathcal{D}$-imersões. Esse trabalho demonstra que essa formulação, de fato, generaliza as imersões suaves. Extensões para outras classes de subvariedades são discutidas através de exemplos e casos particulares.

Palavras–chave

Teoria de Morse. Teoria de Forman. Topologia Computacional. Geometria Computacional. Modelagem Geométrica. Matemática Discreta.

Contents

Quelles sont les trouvailles qui sont des déconvertes et celles qui sont des inventions? La découverte est la trouvaille d'un objet extérieur à nous, qui, même si nous ne l'avons connu que récemment, a toujours existé et existera après nous, et sur lequel notre possibilité de choix est minime. Au contraire, l'invention est la trouvaille d'un objet nouveau, qui n'existait pas avant la trouvaille, et sur laquelle nous avons une grande liberté de choix. [...]
En outre, l'invention des nombres complexes est venue à un certain moment de l'histoire de l'Occident, mais, si cela ne s'était pas produit à ce moment là, elle serait sûrement venue plus tard. Les espaces vectoriels de dimensions ≥ 4, les espaces de Banach et Hilbert, l'inversion et la transformation par polaires réciproques, les distributions, les ondelettes, l'ordinateur, sont des inventions. Mais, une fois trouvés les espaces de Hilbert, la théorie spectrale est une découverte. Une fois trouvées les distributions, leur transformation de Fourier est une découverte. La grotte est une découverte, la hutte une invention, la laine est une découverte, le tissage une invention. On peut jouer longtemps à ce petit jeu de société pas très profond, et tout ce qui vient d'être dit est contestable.
Mais cela montre d'une part que découverte et invention s'entremêlent, qu'elles peuvent être relatives aux objets les plus élementaires comme les plus savants, et qu'il n'y a pas de différence essentielle entre les mathématiques et les autres sciences.

Laurent Schwartz, *Un mathématicien aux prises avec le siècle.*

1
Introduction

Geometry, one of the oldest kinds of sciences, was first recorded in the ancient Mesopotamia when men needed to measure the variations of the tide of a river. It was also used by the Egyptians, who studied shapes, and then first put into an axiomatic form by Euclides in the third century B.C. In the eighteenth century the study of intrinsic structure of geometrical objects made great advances through the work of Euler and Gauß. Gauß' Egregium Theorem states a way for computing the curvature of a surface without considering the ambient space in which the surface lies. In modern terms, this type of surface would be called a manifold. With the emergence of infinitesimal geometry and topology, a common way to describe particular manifolds requires the concept of immersion. In particular we are interested along this work in extending differential tools on immersions to singular geometrical configurations. It is a challenge for mathematicians to study these singular objects. In particular, Geometric Measure Theory (GMT) is a generalization of differential geometry through measure theory. It was mainly created by Federer (Federer 1996, Morgan 2000) to deal with maps and surfaces that are not necessarily smooth. GMT uses tools similar to distribution theory defining rectifiable sets as currents. Integral geometry (Santaló 1953, Langevin 2006) is another way to deduce geometric invariants without differentiation. It has its origin in the theory of geometrical probabilities.

This work is a tentative, among many others to develop a simple formalism to study singular objects. We choose to use distribution theory to extend differential geometry tools, since distributions already extend functions and measures. More specifically, we want to develop a coherent structure that allows substituting classical parameterizations of a differential submanifolds by distributions. Distributions were invented by Schwartz at the end of 1944 to generalize the notion of function. At the time it was a challenge to be able to define the derivative of any function at any point. By generalizing functions with infinitely often differentiable objects, Schwartz' discovery allowed solving many differentials problems, since a distributional derivative always exists, in contrast with the usual derivative. A similar generalization occurred in the history of mathematics when rational numbers were generalized by real numbers in order to solve the square root problem. In that context another generalization eventually emerged later because negative real numbers did not have any polynomial root, the creation of complex numbers solved the problem. The ergonomic aspect of distributions may be the key to unite differential and singular geometry.

Along this work we sketch a formalization mixing distributions and immersions. As distributions are can be infinitely derivable, they are the natural candidates to substitute classical parameterizations. We define $\mathcal{D}$-immersions, a generalization of immersions in the sense that the distribution associated to an immersion is a $\mathcal{D}$-immersion. Besides, we observe that graphs of L^1 functions are $\mathcal{D}$-immersions which motivated us to keep that definition and look further. This would be a first step toward another distributional extension of submanifolds. In particular, we study change of parameterizations through $\mathcal{D}$-immersions in this direction. Similarly, we propose a derivation of tangent cones to the local image of a $\mathcal{D}$-immersion, which match one of the usual tangent cone for images of smooth immersions.

This work is organized as follows. We first introduce basic concepts of differential geometry and distribution theory (Chapters I and II). We then define $\mathcal{D}$-immersions theory (Chapter III) and study some relations with immersions as manifold parameterization (Chapter IV). Finally we show some examples to illustrate this formulation and applications to geometric approximations (Chapter V).

2
Basics of Differential Geometry

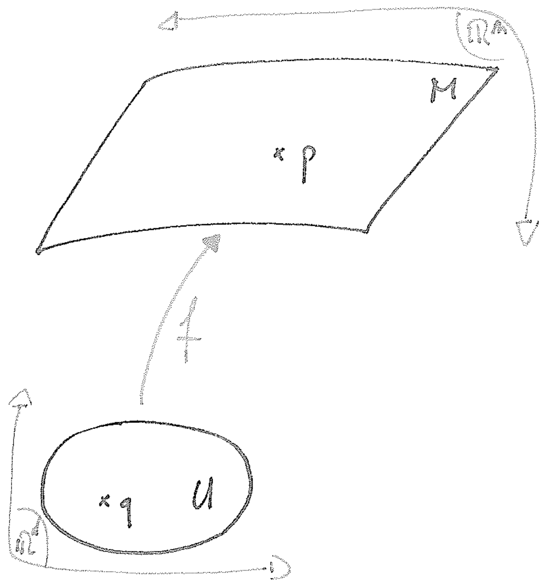


Figure 2.1: An immersion on a submanifold.

In this chapter we define some of the most basic tools of differentiable geometry. We begin by defining immersions to parameterize differential submanifolds. We finally state some useful theorems and definitions that we will need along our work. Along this dissertation we will use the word "smooth" to design C^∞ objects. Further references on differential geometry may be found in (do Carmo 2005).

2.1
Differential immersions and submanifolds

Definition 2.1. *(Immersion) Let U be an open set of $\mathbb{R}^d$. The function $f : U \to \mathbb{R}^n$ is called immersion if for all $x \in U$, f is differentiable and the rank of $D_x f$ is d.*

The notion of submanifold is an important concept in modern geometry since it allows complex structures to be expressed in terms of relatively well-understood properties of simpler spaces such as the Euclidian space $\mathbb{R}^d$. Every point of a submanifold has a neighborhood diffeomorphic to the Euclidian space. It is easier to work on a submanifold than on some unstructured geometrical object. If the local maps are compatible, it is possible to use calculus on a differential submanifold, in particular to define a tangent space.

Definition 2.2. *(Submanifold) $M \in \mathbb{R}^n$ is a* regular submanifold *of dimension* d *if for all $p \in M$ exists a neighborhood $V \subset \mathbb{R}^n$ and a map $f : U \rightarrow V \cap M$ of an open set $U \subset \mathbb{R}^d$ $(d < n)$ onto $V \cap M$ such that:*

- *f is differentiable.*
- *f is a homeomorphism.*
- *$\forall q \in U$, $D_q f : \mathbb{R}^d \rightarrow \mathbb{R}^n$ is one-to-one.*

In such case, f is called a parameterization of M around p.

Remark 2.3. *Every submanifold is locally the image of an immersion (see Figure 2.1).*

This remark is a direct consequence of the definition of a regular submanifold.

Definition 2.4. *(Embedding) Let f be an immersion. If f is a homeomorphism then f is called an embedding.*

Theorem 2.5. *The image of an embedding is a submanifold.*

Theorem 2.6. *Let U be an open set of $\mathbb{R}^d$. Let $f : U \rightarrow \mathbb{R}^n$ be an immersion. For all $x \in U$, there exists a neighborhood W of x in U such that $f|_W : W \rightarrow \mathbb{R}^n$ is an embedding.*

This last theorem permits us to speak about immersed submanifold, i.e. submanifolds which are the images of injective immersions.

2.2 Parameter independence

One of differential geometry main objectives is to study local properties of regular submanifolds. Indeed, according to the definition, local coordinate systems exist in the neighborhood of each point p of a regular submanifold. It is thus possible to define local properties of the submanifold according to these coordinates. For example we can define differentiability at a point $p \in M$. If f is a function from M to another submanifold, an intuitive way to define the differentiability of f on M is to choose a coordinate system in the neighborhood of p and say that f is differentiable on M if its expression in the coordinates of the chosen neighborhood system is a differential map. But a point p of a

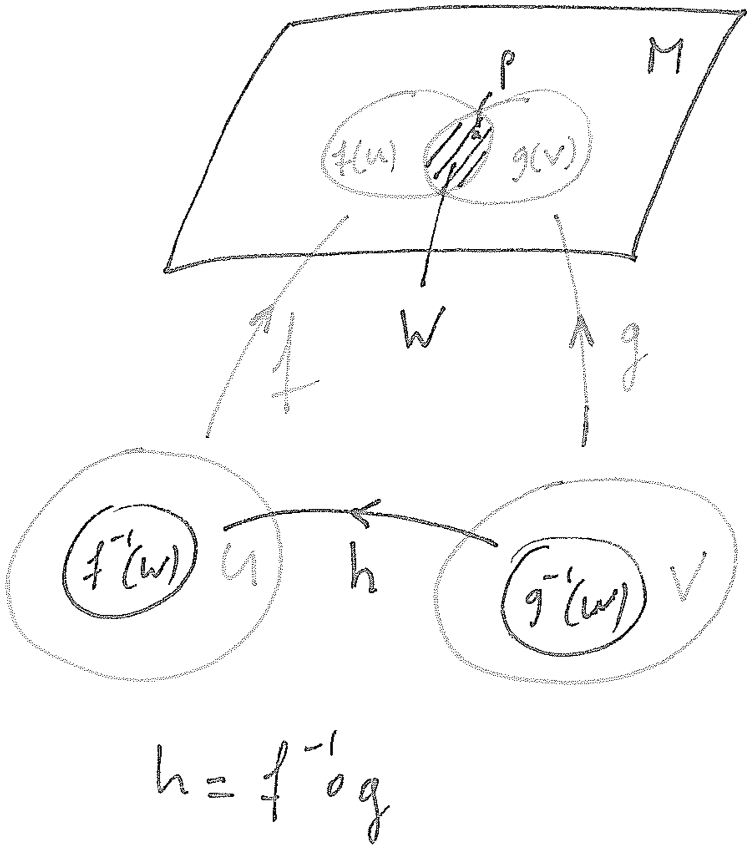


Figure 2.2: Change of parameterization.

regular submanifold belongs to various coordinate neighborhoods and we do not want our definition of differentiability to depend on the chosen coordinate system. Hence in order to define a local property such as differentiability on a regular submanifold geometrically we have to make sure that it is independent of the coordinate system chosen. The next proposition shows that the change of parameters preserves the differential structures of the submanifold.

Proposition 2.7. *Given $p \in M$, M submanifold, let f and g be two parameterizations of M such that:*

$$\begin{cases} f : U \longrightarrow M \\ g : V \longrightarrow M \end{cases} \qquad p \in f(U) \cap g(V) = W.$$

Define $h = (f^{-1} \circ g) : g^{-1}(W) \to f^{-1}(W)$. h is a diffeomorphism (see Figure 2.2).

2.3 Tangent space

Tangent spaces are the best linear approximation of the submanifold. Once tangent spaces have been defined it is possible to define vector fields on

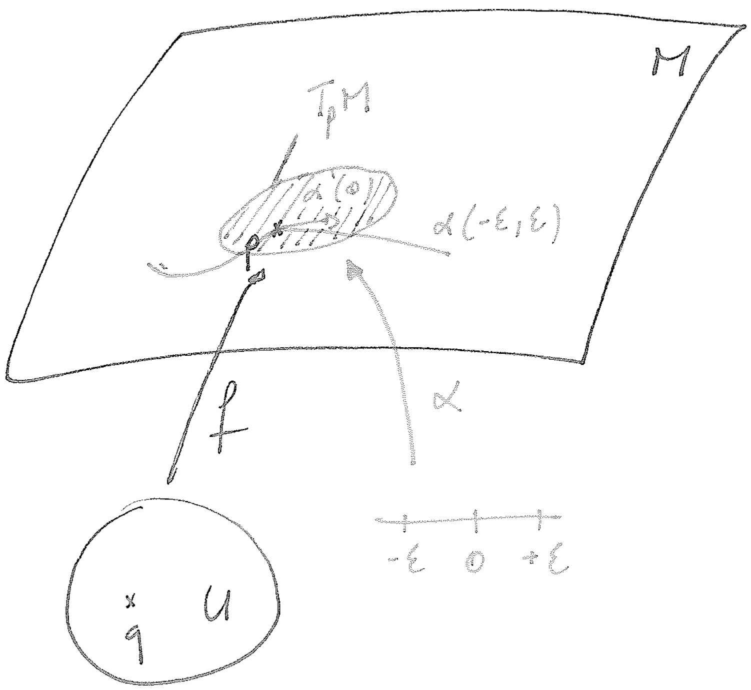


Figure 2.3: Tangent space of a submanifold at a regular point.

a submanifold and, further on, differential equations. The tangent space at a point p of a regular submanifold is a real vector space which intuitively contains all the possible directions in which one can pass through p (see Figure 2.3).

Definition 2.8. *A vector is tangent at p to M if it is the tangent vector $\alpha'(0)$ of a differentiable parameterized curve:*

$$\alpha : (\epsilon; \epsilon) \longrightarrow M \qquad \textit{with} \qquad \alpha(0) = p.$$

In order to define tangent spaces properly, we have to make sure that they do not depend on the parameterization used to parameterize the submanifold, this will ensured by the next proposition.

Proposition 2.9. *Let f be a parameterization of an open set of M and q such that $f(q) = p$.*

The set of tangent vectors at a point $p \in M$ is equal to $D_q f(\mathbb{R}^d)$.

Proof. ($\subset$) If w is a tangent vector at $f(q) = p$ we have $w = \alpha'(0)$ where $\alpha : (-\epsilon; \epsilon) \to M$ and $f(q) = \alpha(0)$.

Define:

$$\beta = f^{-1} \circ \alpha.$$

We have:

$$\beta'(0) = D(f^{-1} \circ \alpha)_0 = (Df)_q^{-1} \circ \alpha'(0).$$

Hence:

$$D_q f(\beta'(0)) = \alpha'(0) = w,$$

and $w \in D_q f(\mathbb{R}^d)$.

($\supset$) If $w = D_q f(v),\ v \in \mathbb{R}^d$. Define: $\gamma(t) = t.v + q$ and $\alpha = f \circ \gamma$, we have:

$$\alpha'(0) = D(f \circ \gamma)_0 = D_q f \circ \gamma'(0) = D_q f(v) = w.$$

Hence w is a tangent vector to the parameterized curve α and therefore is a tangent vector at the point $p \in M$.

□

By the previous proposition, the space $D_q f(\mathbb{R}^d)$ can be calculated from an immersion f, but it does not depend on a particular choice for f. This space will be called the *tangent space* at p to M, and denoted T_pM.

3
Basics of Distribution Theory

To introduce some basics concepts of Distribution theory we first define the convolution product between two functions and then approximations of the identity. Then we define distributions and see how they generalize functions. The last section is dedicated to operations on distributions: the derivative of a distribution and the convolution product of a distribution and function. For further references and proofs see (Lebeau 1999) and see (Schwartz 1997) for more information on the invention of distributions.

3.1
Function approximations

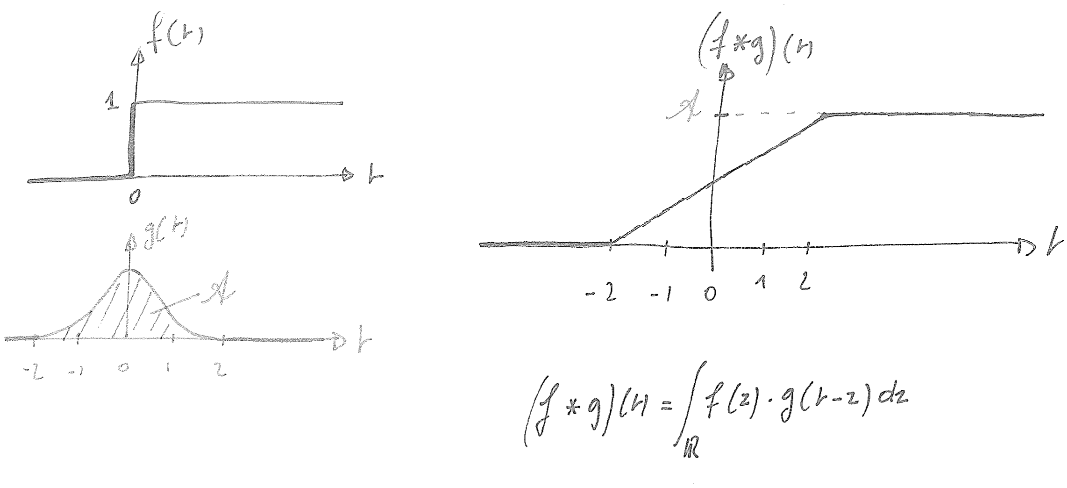


Figure 3.1: Convolution of a discontinuous function f with a smooth test function g.

3.1.1
Convolution

Let $L^1(\mathbb{R}^d)$ denote the normed vector space of integrable functions on $\mathbb{R}^d$, where $\mathbb{R}^d$ is equipped with its Lebesgue measure dx. Let f and g be two function of $L^1(\mathbb{R}^d)$.

Definition 3.1. *(Basic convolution) For $f, g \in L^1(\mathbb{R}^d)$, we call convolution product of f and g, denoted by $f * g$, the element of $L^1(\mathbb{R}^d)$ defined for almost* *every z by:*

$$\big(f * g\big)(x) = \int f(z)g(x-z)dz.$$

Figure 3.1 illustrates the regularization effect of the convolution.

Lemma 3.2. *Let $f \in L^1(\mathbb{R}^d)$, $g \in C^k(\mathbb{R}^d)$. If g admits limited partial derivatives $\partial^\alpha g$ for all multi-indexes α, $|\alpha| \leq k$, then $f * g \in C^k(\mathbb{R}^d)$ and for $|\alpha| \leq k$ we have:*

$$\partial^\alpha (f * g) = f * \partial^\alpha g.$$

3.1.2
Regularization

Let φ be a C^∞ real-valued function with support in the ball $\{\|t\| \leq 1\}$ whose integral is equal to 1:

$$\int_{\mathbb{R}^d} \varphi(t)dt = 1.$$

Definition 3.3. *(Approximation of the identity) We call approximation of the identity the family of functions*

$$\Big(\varphi_\epsilon : t \longmapsto \epsilon^{-d}\varphi(t/\epsilon) \quad , \quad 0 < \epsilon \leq 1\Big).$$

Note that the φ_ϵs are C^∞ functions with support in the ball $\{\|t\| \leq \epsilon\}$ and of integral equals to 1, since

$$\int_{\mathbb{R}^d} \epsilon^{-d}\varphi(t/\epsilon)dt = \int_{\mathbb{R}^d} \varphi(t)dt = 1.$$

Lemma 3.4. *Let f be a continuous function with compact support on $\mathbb{R}^d$. The functions $f_\epsilon = f * \varphi_\epsilon$ belong to $C^\infty(\mathbb{R}^d)$, have a compact support and converge uniformly on $\mathbb{R}^d$ to f when ϵ tends to 0.*

Theorem 3.5. *For all $f \in L^1(\mathbb{R}^d)$, the functions $f_\epsilon = f * \varphi_\epsilon$ belong to $L^1 \cap C^\infty$, and converge in the L^1 norm to f when ϵ tends to 0.*

$$(f * \varphi_\epsilon) \underset{\epsilon \to 0}{\longrightarrow} f \quad in \quad L^1.$$

In particular, the space of $C^\infty(\mathbb{R}^d)$ functions with compact support is dense in $L^1(\mathbb{R}^d)$.

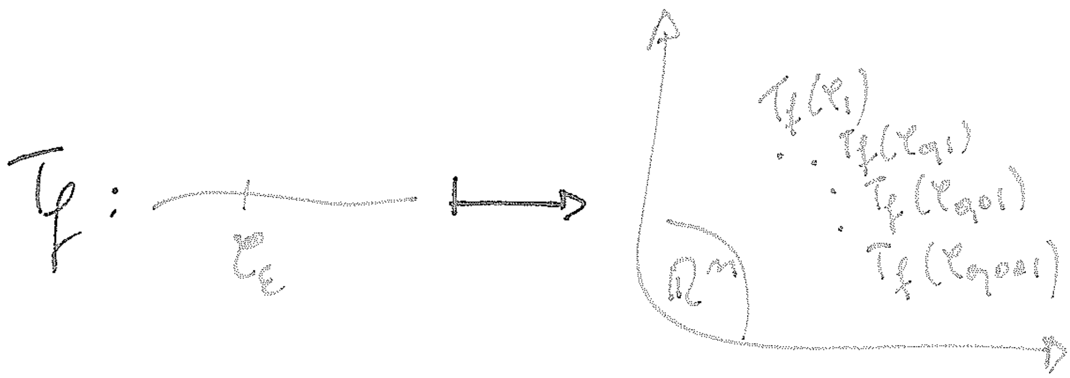


Figure 3.2: A distribution associates a point to a test function.

3.2 Distributions

For K compact of $\mathbb{R}^d$, we denote by C_K^∞ the space of C^∞ functions of $\mathbb{R}^n$ with support included in K. Let Ω be an open of $\mathbb{R}^d$. $C_0^\infty(\Omega)$ is the union of C_K^∞'s where K is a compact in Ω. The elements of $C_0^\infty(\Omega)$, i.e. infinitely often differentiable functions with compact support in Ω, are called *test functions.*

Definition 3.6. *(Distributions) A distribution on Ω is a linear form T of $C_0^\infty(\Omega)$*

$$\varphi \longmapsto \langle T, \varphi\rangle \in \mathbb{R} \qquad \varphi \in C_0^\infty(\Omega).$$

which satisfies the following property: for all compact K in Ω, there exists an integer p and a constant C such that

$$\forall \varphi \in C_K^\infty \qquad |\langle T, \varphi\rangle| \leq C \sup_{\substack{|\alpha|\leq p \\ x\in K}} |\partial^\alpha \varphi(t)|. \tag{3-1}$$

We denote by $\mathcal{D}'(\Omega)$ the space of distributions on Ω (see Figure 3.2). It is a vector space. When the integer p can be chosen independently from K, we say that the order of the distribution T is finite, and the smallest possible value of p is called the order of T.

Distribution are "generalized functions". Let $L^1_{loc}(\Omega)$ be the space of functions locally integrable on Ω. An element of $L^1_{loc}(\Omega)$ is the data of a Lebesgue-measurable function f on Ω, satisfying $\int_K |f(x)|dx < \infty$ for all compact $K \in \Omega$: two such functions are identified if and only if $f(x) = g(x)$ almost everywhere. We write T_f, the distribution associated to an element $f \in L^1_{loc}(\Omega)$ i.e.

$$\langle T_f, \varphi\rangle = \int_\Omega f(x)\varphi(x)dx \qquad \forall \varphi \in C_K^\infty.$$

From definition 3.6, we have $|\langle T_f, \varphi\rangle| \leq C\sup_{x\in K}|\varphi(x)|$ with $C = \displaystyle\int_K |f(x)|dx$, so the regularity condition (3-1) is satisfied for $p = 0$. The next lemma identifies $L^1_{loc}(\Omega)$ with a subspace of $\mathcal{D}'(\Omega)$.

Lemma 3.7. *Let f and g be two functions locally integrable on Ω. The following properties are equivalent:*

– $f(x) = g(x)$ *almost everywhere.*
– $\displaystyle\int f(x)\varphi(x)dx = \int g(x)\varphi(x)dx$ *for all* $\varphi \in C_0^\infty(\Omega)$, *i.e.* $T_f = T_g$.

Example 3.8. *Dirac's distribution at $a \in \mathbb{R}^d$, δ_a is defined by*

$$\langle \delta_a, \varphi \rangle = \varphi(a).$$

It is a distribution of order 0 *on $\mathbb{R}^d$. If $\chi_\epsilon(x) = \epsilon^{-1}\chi(x/\epsilon)$ is an approximation of the identity, the χ_ϵ's converge point-wise in $\mathcal{D}'(\mathbb{R}^d)$ to δ_0 since*

$$\langle \chi_\epsilon, \varphi \rangle = \int \varphi(x)\chi_\epsilon(x)dx = \big(\varphi * \check{\chi}_\epsilon\big)(0).$$

where $\check{\chi}_\epsilon(x) = \chi_\epsilon(-x) = \epsilon^{-1}\chi(-x/\epsilon)$ is also an approximation of the identity, hence by the lemma 3.4:

$$\lim_{\epsilon \to 0} \langle \chi_\epsilon, \varphi \rangle = \varphi(0) = \langle \delta_0, \varphi \rangle .$$

Example 3.9. *Another example of distribution, but of order* > 0, *is d_a defined for $a \in \mathbb{R}$ by*

$$\langle d_a, \varphi \rangle = \varphi'(a).$$

3.3
Operations on distributions

Definition 3.10. *(Derivation) The partial derivatives $\dfrac{\partial T}{\partial x_i}$ of a distribution $T \in \mathcal{D}'(\Omega)$ are the distributions on Ω defined by:*

$$\left\langle \frac{\partial T}{\partial x_i}, \varphi \right\rangle = \left\langle T, -\frac{\partial \varphi}{\partial x_i} \right\rangle \qquad \forall \varphi \in C_0^\infty(\Omega).$$

Hence for $T \in D'(\Omega), \varphi \in C_0^\infty(\Omega)$

$$\left\langle \frac{\partial}{\partial x_i}\left(\frac{\partial T}{\partial x_j}\right), \varphi \right\rangle = \left\langle \frac{\partial T}{\partial x_j}, \frac{\partial \varphi}{\partial x_i} \right\rangle = \left\langle T, \frac{\partial^2 \varphi}{\partial x_i \partial x_j} \right\rangle .$$

By Schwartz's lemma: $\dfrac{\partial}{\partial x_i}\left(\dfrac{\partial T}{\partial x_j}\right) = \dfrac{\partial}{\partial x_j}\left(\dfrac{\partial T}{\partial x_i}\right)$. Therefore the order of derivation does not affect the result of a successive derivation of a distribution, and for $\alpha \in \mathbb{N}^n$ multi-index, we have:

$$\langle \partial^\alpha T, \varphi \rangle = (-1)^{|\alpha|} \langle T, \partial^\alpha \varphi \rangle .$$

Besides, if $T = f(x)$ is a C^1 function, Definition 3.10 means that for all $\varphi \in C_0^\infty$ we have:

$$\left\langle \frac{\partial T}{\partial x_j}, \varphi \right\rangle = \left\langle T, \frac{\partial \varphi}{\partial x_j} \right\rangle = -\int f(x) \frac{\partial \varphi}{\partial x_j}(x) dx = \int \frac{\partial f}{\partial x_j}(x) \varphi(x) dx.$$

Hence the distributional derivative, $\dfrac{\partial T_f}{\partial x_j}$ of T_f is the distribution associated to the usual derivative $\dfrac{\partial f}{\partial x_j}(x)$ for f of class C^1: $\frac{\partial T_f}{\partial x_j} = \frac{T \cdot \partial f}{\partial x_j}$.

Example 3.11. *Let $H(t)$ be Heaviside's function defined for $t \in \mathbb{R}$ by*

$$\begin{cases} H(t) = 1 & for\ t \geq 1 \\ H(t) = 0 & for\ t < 0 \end{cases}$$

$H \in L^1_{loc}(\mathbb{R})$ and is thus associated to a distribution. Computing its distributional derivative we obtain:

$$\langle T'_H, \varphi \rangle = -\langle T_H, \varphi' \rangle = -\int_0^\infty \varphi'(t) dt = \varphi(0) = \langle \delta_0, \varphi \rangle .$$

Hence:

$$T'_H = \delta_0.$$

Example 3.12. *Let $\delta_a(t)$ be Dirac's distribution defined for $t \in \mathbb{R}$. We can check that $\delta'_a = -d_a$:*

$$\langle \delta'_a, \varphi \rangle = -\langle \delta_a, \varphi' \rangle = -\varphi'(a).$$

Theorem and Definition 3.13. *(Substitution formula) Let Ω_1 and Ω_2 be two open subsets of $\mathbb{R}^d$ and $\phi : \Omega_1 \to \Omega_2$ a C^∞ diffeomorphism. For $T \in D'(\Omega_2)$, the formula:*

$$\forall \varphi \in C_0^\infty(\Omega_1), \quad \langle T \circ \phi, \varphi \rangle = \langle T, \psi \rangle \quad with \quad \psi(y) = \frac{\varphi\big(\phi^{-1}(y)\big)}{|\det J\big(\phi^{-1}(y)\big)|}.$$

defines a distribution on Ω_1, called inverse image of T by the change of parameter ϕ.

The above definitions match the usual formula for distributions associated to a function in L^1.

Now, we aim at calculating the convolution product of distributions, we first define the convolution product of a distribution and a test function. Let $T \in \mathcal{D}'(\mathbb{R}^d)$ and $\varphi \in C_0^\infty(\mathbb{R}^d)$. We define a function of x by setting:

$$\big(T * \varphi\big)(x) = \langle T, \tau_x \check{\varphi} \rangle \,. \tag{3-2}$$

where $\tau_x \check{\varphi} \in C_0^\infty(\mathbb{R}^d)$ is the function $z \mapsto \check{\varphi}(z - x) = \varphi(x - z)$. For $T = f(x) \in L^1_{loc}$, the formula above is equivalent to the usual definition $\big(f * \varphi\big)(x) = \displaystyle\int f(z)\varphi(x - z)dz$. Observe that $T * \varphi$ is always a C^∞ function.

Proposition 3.14. *We denote by $T * \varphi$ the convolution product of T and φ. It is a C^∞ function on $\mathbb{R}^d$ satisfying for all α:*

$$\partial^\alpha (T * \varphi) = T * \partial^\alpha \varphi.$$

Note that approximations of the identity also regularize distributions:

Proposition 3.15. *If φ_ϵ is an approximation of the identity, then we have the convergence in D': $T * \varphi_\epsilon \underset{\epsilon \to 0}{\longrightarrow} T$, i.e.:*

$$\forall \phi \in C_0^\infty(\Omega_1), \langle T * \varphi_\epsilon, \phi \rangle = \langle T, \phi * \check{\varphi}_\epsilon \rangle \underset{\epsilon \to 0}{\longrightarrow} \langle T, \phi \rangle \;\; in \; \mathbb{R}.$$

4
Building Immersions with Distributions

The main objective of this work is to use distribution derivation on non-smooth immersions. Distributions are infinitely often differentiable objects, similarly to smooth parameterization. Therefore they naturally extend class conditions on immersions. In this chapter we set up a formulation for $\mathcal{D}$-immersions trying to preserve the main geometric properties of immersions.

4.1
Brute $\mathcal{D}$-parameterization: a first attempt

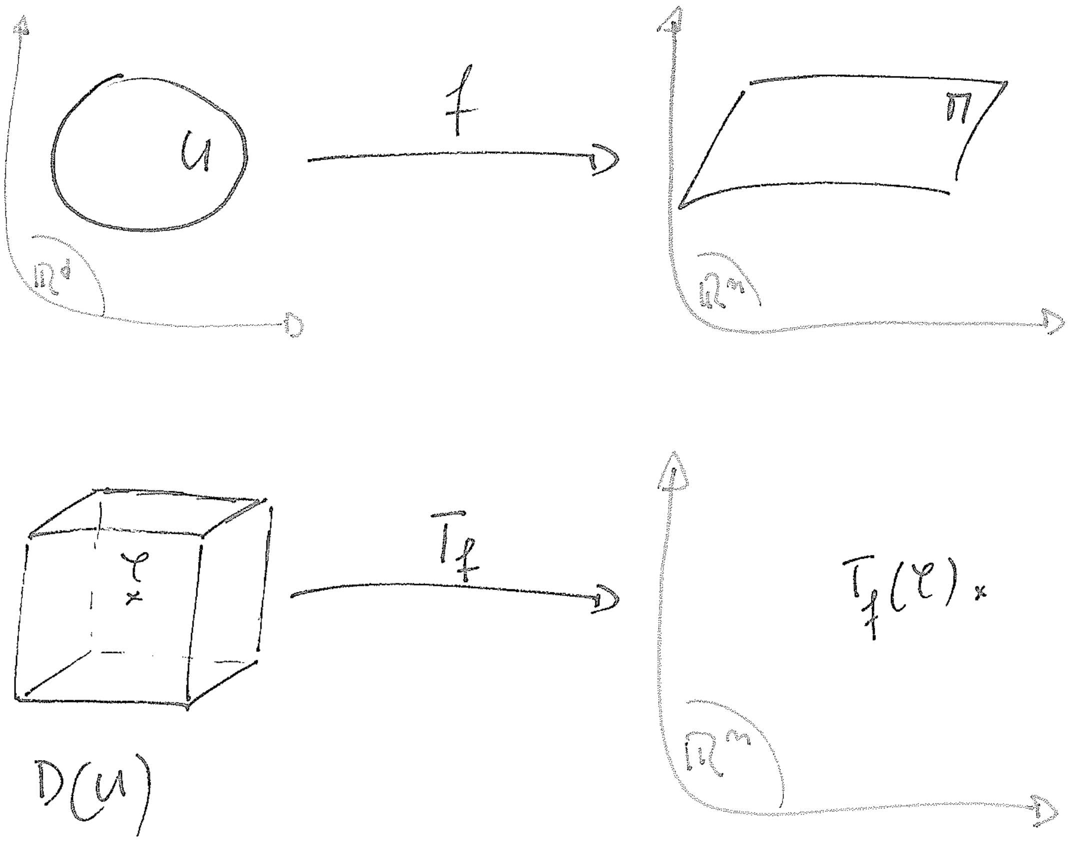


Figure 4.1: Parameterization directly from $D(\mathbb{R}^d)$.

There is a very direct way to substitute classical parameterizations by distributions. Distributions are defined on test functions spaces. Therefore, if we use a distribution T instead of a parameterization f, we change the parameter space from a subset U of $\mathbb{R}^d$ to $D(U)$. However test functions space $D(U)$ have infinite dimensions, and thus the submanifold parameterized on a test function space could have as many dimensions as the co-domain of the distribution used has (see Figure 4.1). This is clearly an undesirable fact. Another drawback concerns the derivative of our parameterization; we do not know how to interpret, in terms of tangent space, the derivation of T with respect to the space of test functions: $D_{\varphi_0}(T)$. As a direct use of distributions as parameterizations may not work mainly because of the non-finite dimension of the parameter space, we propose to structure differently the parameter spaces, in an approximation perspective.

4.2 Approximating by convolution

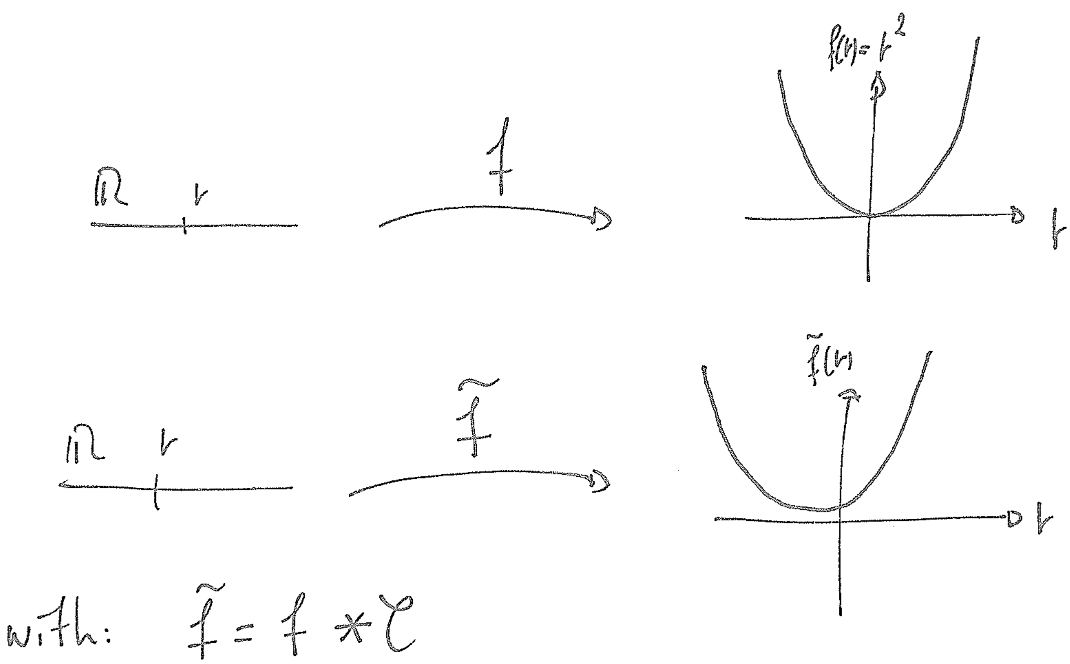


Figure 4.2: Convolution regularizes the parameterization.

On the one hand, defining differential properties from non-differential objects is often handled as an approximation problem, usually requiring convolution operations. On the other hand we saw in the previous chapter that applying a distribution and convolution product are related operations and that we could switch from one to another easily. Our approach lies in the regularization properties of the convolution product. Indeed, when computing the convolution product of a distribution with a test function, we obtain a C^∞ function (see Figure 4.2). Hence, convolution allows using distributions as smooth parameterizations. Moreover, it is possible to think of a more general parameterization not relying on an existing embedding: by taking an arbitrary distribution we can always generate a new parameterization by computing its convolution product with a test function. More specifically, let f be an immersion on $U \subset \mathbb{R}^d$, parameterizing its image $f(U) = M$ (M is a submanifold of dimension d in $\mathbb{R}^n$). Let T_f be the distribution associated to f, and φ be a test function on U. We define the function $\tilde{f}$ by:

$$\tilde{f} : \begin{cases} U & \longrightarrow \widetilde{M} \subset \mathbb{R}^n \\ x & \longmapsto \big(T_f * \varphi\big)(x) \end{cases}$$

Since the convolution averages functions, the function $\tilde{f}$ does not map U exactly on M, the image of the immersion f. Hence $\tilde{f}$ maps U on $\widetilde{M} = \tilde{f}(U)$, a mean set of M in $\mathbb{R}^n$. Hence $\tilde{f}$ parameterizes a geometrical object that corresponds to means of a classical submanifold $M = f(U)$ and those means depend on the test function φ used. In order to parameterize the original M, we need to choose the test function φ such that $\widetilde{M} = M$. This is in general not possible directly, but at the limit as in the regularization seen in Section 3.1.2. Formally, we can define a sequence of φ_ϵ of test functions such that $\big(T_f * \varphi_\epsilon\big)(U) \underset{\epsilon \to 0}{\longrightarrow} M$.

4.3
$\mathcal{D}$-immersions

In the formalization of this approach, we will try to preserve the geometric properties of the immersion f. We called the equivalent formulation for immersion $\mathcal{D}$-immersions (see Figure 4.3):

Definition 4.1. *($\mathcal{D}$-immersions) T is a $\mathcal{D}$-immersion if for all approximation of the identity φ_ϵ there exists $\epsilon_0 > 0$ such that for all $\epsilon \in]0, \epsilon_0[$, $T * \varphi_\epsilon$ is an immersion.*

To ensure that $\mathcal{D}$-immersions actually generalize classical immersions in the same way distributions generalize functions, we have to make sure that the distribution associated to a smooth immersion is actually a $\mathcal{D}$-immersion. This is done in the following theorem.

Theorem 4.2. *Let f be an immersion such that $f : U \to M$, where U is an open set of $\mathbb{R}^d$ such that $\overline{U}$ contains no singularity of f. If T_f is the distribution associated to f then T_f is a $\mathcal{D}$-immersion.*

Proof. Let φ_ϵ be an approximation of the identity, define f_ϵ such that:

$$f_\epsilon(x) = \big(T_f * \varphi_\epsilon\big)(x) = \int_U f(z) \cdot \varphi_\epsilon(x - z) dz = \big(f * \varphi_\epsilon\big)(x).$$

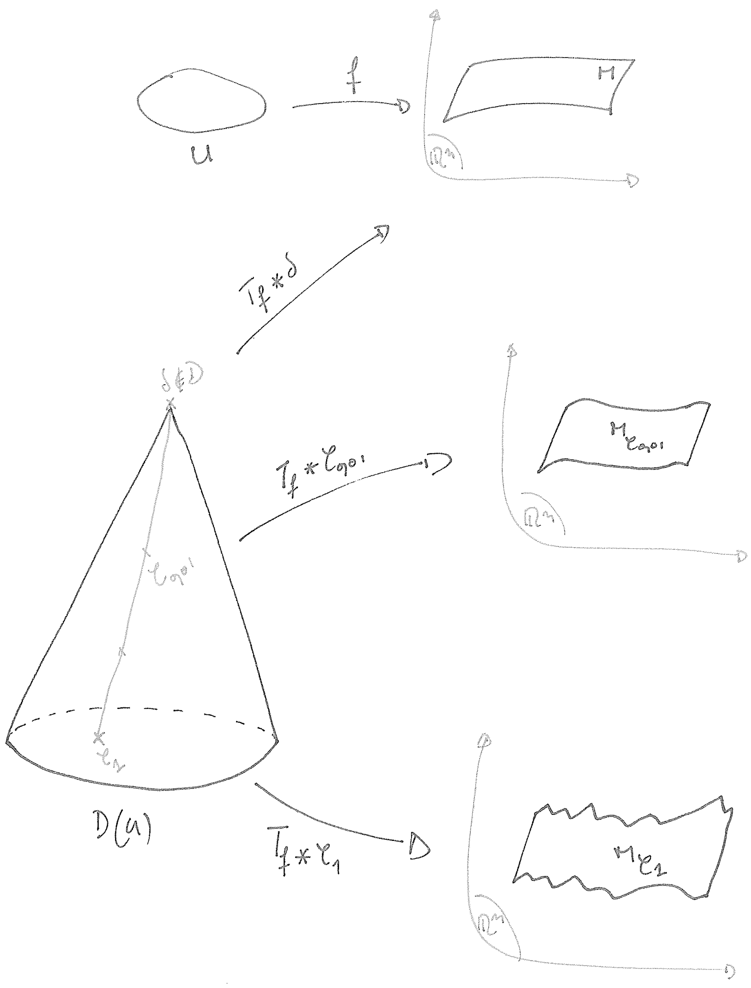


Figure 4.3: Image of a $\mathcal{D}$-immersion associated to an immersion.

We have to prove that for ϵ small enough f_ϵ is an immersion. For that, we will show that $D_x f_\epsilon$ has maximal rank for all $x \in U$ and $0 < \epsilon < \epsilon_0$. First observe that $D_x f_\epsilon$ is actually an approximation of $D_x f$:

$$D_x f_\epsilon : \begin{cases} \mathbb{R}^d & \longrightarrow \mathbb{R}^n \\ x & \longmapsto \big(Df * \varphi_\epsilon\big)(x) \end{cases}.$$

Since f is an immersion, $D_x f$ has at least one non-vanishing minor, i.e. there exists a $d \times d$ matrix $[D_x f]_d$ extracted from $D_x f$ such that $\det[D_x f]_d \neq 0$ for all $x \in U$. Denote by $[D_x f_\epsilon]_d$ the matrix extracted from $D_x f_\epsilon$ in the same way that $[D_x f]_d$ is extracted from $D_x f$.

The smaller ϵ, the closer $\det[D_x f_\epsilon]_d$ is from $\det[D_x f]_d$ and the further from 0. Formally, Theorem 3.5 ensures that $\det[D_x f_\epsilon]_d \underset{\epsilon \to 0}{\longrightarrow} \det[D_x f]_d$, since det is a continuous function:

$$\forall \alpha > 0,\ \exists \beta_\alpha > 0 \text{ such that } |\epsilon - 0| < \beta_\alpha \Rightarrow |\det[D_x f_\epsilon]_d - \det[D_x f]_d| < \alpha.$$

Choose α_0 to be:

$$\alpha_0 = \inf_{x \in U} \big(|\det[D_x f]_d|\big).$$

Since there are no singularities in $\overline{U}$, $\alpha_0 > 0$.
Since a ball centered in $\det[D_x f]_d$ of radius inferior to α_0 on the real line does not contain 0, for all $|\epsilon| < \beta_{\alpha_0}$ we have that $\det[D_x f_\epsilon]_d \neq 0$ and thus $D_x f_\epsilon$ has maximal rank. Concluding for all $x \in U$, exists $\beta_{\alpha_0} > 0$ such that for all $\epsilon < \beta_{\alpha_0}$, f_ϵ is an immersion. □

4.4
Graph of a function: $\mathcal{D}$-immersions from non-smooth immersions

In this section, we will prove that parameterizations of graph of functions, even if only L^1, are associated to $\mathcal{D}$-immersions. This allows using $\mathcal{D}$-immersions for a much wider class of objects.

Theorem 4.3. *Let $M \in \mathbb{R}^n$ be the graph of a function $u \in L^1 : \mathbb{R}^{n-1} \to \mathbb{R}$. Let U be an open of $\mathbb{R}^{n-1}$ and f be the parameterization of M such that:*

$$f : \begin{cases} U & \longrightarrow M \\ x & \longmapsto \big(f_1(x), \ldots, f_n(x)\big) \end{cases}$$

Where $\forall i \in \{1, \ldots, n-1\},\ f_i : x \in \mathbb{R}^{n-1} \mapsto x_i \in \mathbb{R}$ and $f_n(x) = u(x)$.
The distribution associated to f is a $\mathcal{D}$-immersion.

Proof. Let φ_ϵ be an arbitrary approximation of the identity. We have to prove that the mean map of f is an immersion. Defines f_ϵ as being the mean map of f, $f_\epsilon = T_f * \varphi_\epsilon = f * \varphi_\epsilon$:

$$f_\epsilon \begin{cases} U & \longrightarrow M_{\varphi_\epsilon} \\ x & \longmapsto \Big(\big(f_1 * \varphi_\epsilon\big)(x), \ldots, \big(f_n * \varphi_\epsilon\big)(x)\Big) \end{cases}$$

We have to show that the rank of the Jacobian matrix is $n-1$. The Jacobian matrix of f_ϵ is:

$$\begin{pmatrix} \frac{\partial(f_1 * \varphi_\epsilon)}{\partial x_1} & \cdots & \frac{\partial(f_1 * \varphi_\epsilon)}{\partial x_{n-1}} \\ \vdots & & \vdots \\ \frac{\partial(f_n * \varphi_\epsilon)}{\partial x_1} & \cdots & \frac{\partial(f_n * \varphi_\epsilon)}{\partial x_{n-1}} \end{pmatrix}_{n\times\ n-1} .$$

$$\text{Since} \qquad \forall i \in \{1, \ldots, n-1\} \quad f_i(x) = x_i$$

$$\text{we have} \quad \frac{\partial(f_i * \varphi_\epsilon)}{\partial x_j} = \frac{\partial(f_i)}{\partial x_j} * \varphi_\epsilon = \frac{\partial(x_i)}{\partial x_j} * \varphi_\epsilon.$$

$$\frac{\partial(f_i * \varphi_\epsilon)}{\partial x_j} = \begin{cases} 1 * \varphi_\epsilon = \displaystyle\int 1 \cdot \varphi_\epsilon(x) dx = 1 & if \quad i = j \\ 0 * \varphi_\epsilon = 0 & if \quad i \neq j \end{cases} \qquad \forall (i,j) \in \{1, \ldots, n-1\}.$$

Thus the Jacobian matrix of f_ϵ is:

$$\begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & 0 & \ddots & 0 & \vdots \\ 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & \dots & 0 & 1 \\ f_n * \frac{\partial(\varphi_\epsilon)}{\partial x_1} & & \dots & & f_n * \frac{\partial(\varphi_\epsilon)}{\partial x_{n-1}} \end{pmatrix}_{n\times\ n-1}.$$

Hence the rank of the Jacobian matrix of f_ϵ is $n-1$, and consequently f_ϵ is an immersion. □

5
Toward Geometric Properties of $\mathcal{D}$-Immersions

In the previous chapter we managed to generalize the concept of immersion by defining $\mathcal{D}$-immersion. The choice of approximations of the identity as test functions allows recovering of original parameterizations when parameter ϵ tends to 0, in the case of a $\mathcal{D}$-immersion associated with a smooth immersion. We will now exhibit geometric properties of $\mathcal{D}$-immersions. We want to know what kind of structure is mapped through a $\mathcal{D}$-immersion, and we will thus focus on a possible structuration of $D(U)$.

5.1
Structuring $D(U)$

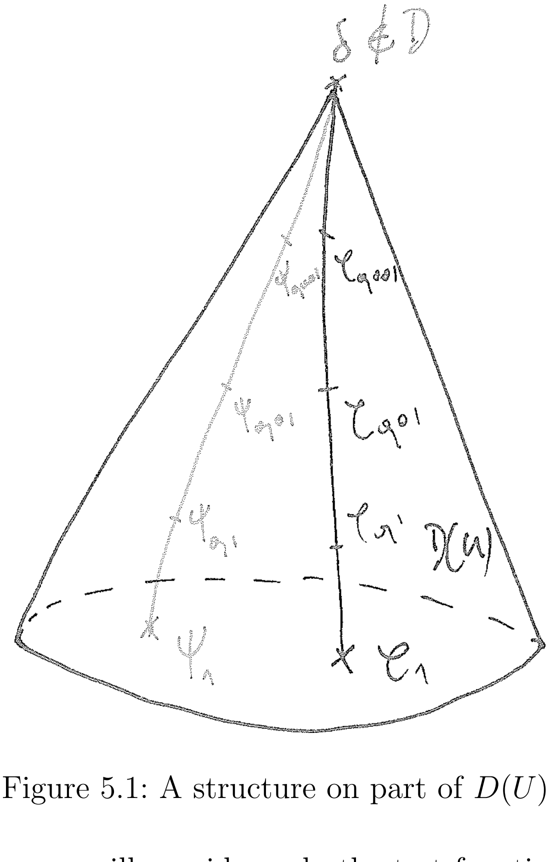


Figure 5.1: A structure on part of $D(U)$.

In this section, we will consider only the test functions $\phi \in D(\mathbb{R}^d)$ with unit mass, denoting

$$D_*(U) = \{\phi \in D(U), \int_U \phi = 1\} \quad .$$

An immersion maps the local structure of $\mathbb{R}^d$ onto its image. Similarly, a $\mathcal{D}$-immersion maps a certain structure of $D_*(\mathbb{R}^d)$ onto its image. This structure is a collection of approximation curves: any $\phi \in D_*(\mathbb{R}^d)$ can be seen as an approximation of the identity $\phi = \tau_z \check{\varphi}_\epsilon$ for a certain z and ϵ (see Figure 5.1). For fixed ϵ, varying z, the test functions $\tau_z \check{\varphi}_\epsilon$ span a d-dimensional object in $D_*(\mathbb{R}^d)$. Reducing ϵ generates approximations of the identity at each point of this object. A $\mathcal{D}$-immersion maps this structure in $\mathbb{R}^n$: given a $\mathcal{D}$-immersion T, the image $\langle T, \phi \rangle \in \mathbb{R}^n$ of an arbitrary test function with unit mass ϕ is mapped onto an object $M_{\varphi_\epsilon} = T * \varphi_\epsilon(U)$ by $T * \varphi_\epsilon(z) = \langle T, \tau_z \check{\varphi}_\epsilon \rangle = \langle T, \phi \rangle$. If ϵ is small enough, but not zero, M_{φ_ϵ} is a smooth d-submanifold. Intuitively, part of the image of a $\mathcal{D}$-immersion can be seen as a collection of smooth submanifold in $\mathbb{R}^n$, eventually tending to an object in $\mathbb{R}^n$.

5.2 Compatible $\mathcal{D}$-immersions: change of parameters

In the classical context, we are able to characterize if two immersions define the same object. More precisely: $f : U_f \to M$ and $g : U_g \to M$ define the same object if $h = g^{-1} \circ f$ is a diffeomorphism. Applying h is called a change of parameters, and two immersions are compatible if they locally define the same object. Properties derived from immersions are geometric if they are invariant by change of parameters, otherwise they are merely analytical. We would like to state a similar characterization for $\mathcal{D}$-immersions. Since we use distributions as parameterizations, the change of parameters has to be done in spaces of test functions (see Figure 5.2).

Definition 5.1. *($\mathcal{D}$-change of parameters) We say that two $\mathcal{D}$-immersions $T \in D'(U)$ and $S \in D'(V)$ define the same object if:*

$$\exists h : U \to V \text{ and } \forall \varphi \in D_*(U),\ \exists \epsilon_0 > 0 \text{ and } \psi \in D_*(V) \quad \text{such that}: \\ \forall \epsilon \in]0, \epsilon_0[, \forall u \in U, (T * \varphi_\epsilon)(u) = (S * \psi_\epsilon)(h(u)) \quad .$$

This definition may be restrictive, but it extends the notion of compatible immersions at least in the linear case, as stated in the next lemma:

Lemma 5.2. *If f and g are two compatible C^∞-immersions and $h = g^{-1} \circ f$ is linear, then T_f and T_g are compatible $\mathcal{D}$-immersions.*

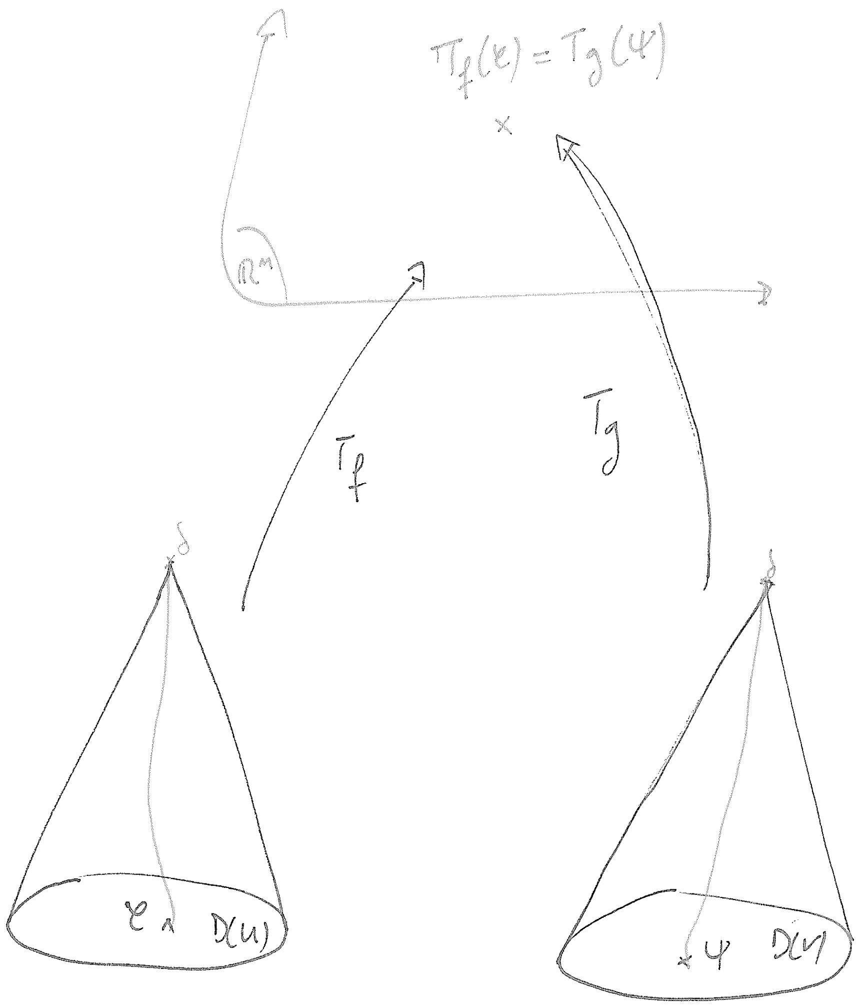


Figure 5.2: Compatible $\mathcal{D}$-immersions.

Proof. We know from Theorem 4.2 that T_f and T_g are $\mathcal{D}$-immersions. We have to check if they locally define the same object. Let $h = g^{-1} \circ f$ be the diffeomorphism mapping the domains U and V of f and g respectively.
Given $\varphi \in D_*(U)$, we want to determine $\psi \in D_*(V)$ such that $\langle T_f, \tau_u \check{\varphi}_\epsilon \rangle = \left\langle T_g, \tau_{h(u)} \check{\psi}_\epsilon \right\rangle$. We have:

$$
\begin{aligned}
\langle T_f, \tau_u \check{\varphi}_\epsilon \rangle &= \int_U f(x) \cdot \varphi_\epsilon(u - x) dx \\
&= \int_U g \circ g^{-1} \circ f(x) \cdot \frac{1}{\epsilon^d} \varphi \left(\frac{u - x}{\epsilon} \right) dx \\
&= \int_U g \circ h(x) \cdot \frac{1}{\epsilon^d} \varphi \left(\frac{u - x}{\epsilon} \right) dx \\
&= \int_V g \circ h\left(h^{-1}(y)\right) \cdot \frac{1}{\epsilon^d} \varphi \left(\frac{u - h^{-1}(y)}{\epsilon} \right) \cdot \left| \det J\left(h^{-1}\right) \right|^{-1} (y)\, dy \\
&= \int_V g(y) \cdot \varphi_\epsilon \left(u - h^{-1}(y)\right) \cdot \left| \det J\left(h^{-1}\right) \right|^{-1} (y)\, dy \quad .
\end{aligned}
$$

We can define, for a given ϵ:

$$\psi(y) = \varphi\left(h^{-1}(y)\right) \cdot \left|\det J\left(h^{-1}\right)\right|^{-1}(y)\,.$$

Observe that since h is a C^∞ diffeomorphism, ψ is C^∞ with support in V. Moreover, since we supposed that h is linear, we have that $J\left(h^{-1}\right)$ is a constant matrix and

$$\begin{aligned}\tau_{h(u)}\check{\psi}_\epsilon(y) &= \frac{1}{\epsilon^d}\,\psi\left(\frac{h(u)-y}{\epsilon}\right) = \frac{1}{\epsilon^d}\,\varphi\left(h^{-1}\left(\frac{h(u)-y}{\epsilon}\right)\right) \cdot \left|\det J\left(h^{-1}\right)\right|^{-1} \\ &= \varphi_\epsilon\left(u - h^{-1}(y)\right) \cdot \left|\det J\left(h^{-1}\right)\right|^{-1} \quad .\end{aligned}$$

Finally, $T_f * \varphi_\epsilon(u) = \langle T_f, \tau_u \check{\varphi}_\epsilon\rangle = \int_V g(y) \cdot \tau_{h(u)}\check{\psi}_\epsilon(y)dy = \left\langle T_g, \tau_{h(u)}\check{\psi}_\epsilon\right\rangle = T_g * \psi_\epsilon(h(u))$, with $\psi \in D_*(V)$. □

From the last observation of the proof, the change of parameters works efficiently for the C^∞ case with linear domain mapping, but unfortunately the substitution formula does not work directly for other classes of functions. Here we face a delicate point of our proposal if we want to extend differential tools to the C^0 case.

5.3
The C^1 case

We were not able to define $\mathcal{D}$-change of parameters that extend directly C^0-subsitutions. However, it should be possible in the C^1 case. Indeed, we conjecture the $\mathcal{D}$-immersions associated to compatible C^1 immersions are $\mathcal{D}$-compatible. Follow elements of an eventual proof. Let f and g be two C^1 embeddings and name T_f and T_g their associated distributions. Given $\varphi \in C_0^\infty$ a test function, the C^1 substitution is $h = g^{-1} \circ f$. By the substitution formula of Theorem 5.2, we obtain $\psi = \varphi \circ h^{-1} |\det J(h^{-1})|^{-1}$. Since h is only C^1, ψ is only locally C^1. Hence ψ is not a test function. Given $\delta > 0$, there exists $\tilde{\psi}_\delta \in C_0^\infty$ such that:

$$\left|\int g \cdot \psi - \left\langle T_g, \tilde{\psi}_\delta \right\rangle\right| < \delta.$$

In that case, $\tilde{\psi}_\delta$ is a test function that may approximate the desired change of parameters for the C^1 case.

5.4
Tangent cones from $\mathcal{D}$-immersions

In order to study singular objects we have to be able to define approximations spaces upon singularities. Since tangent spaces cannot be defined everywhere on singular objects we propose a definition of tangent cone. We will see that this definition matches the definition of the common tangent space on regular objects.

5.4.1
Regular case

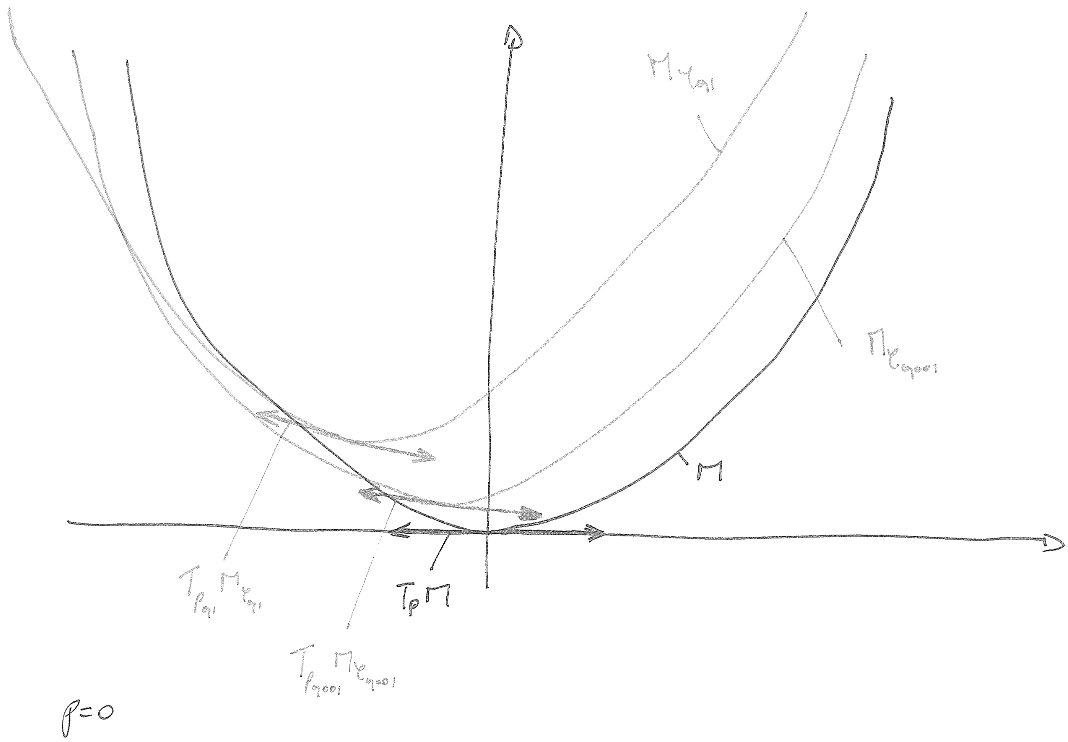


Figure 5.3: The tangent space at the images of the mean maps.

Consider a smooth immersion f and its associated $\mathcal{D}$-immersion T_f, and a fixed parameter $q \in U$. As recalled in Section 2.3, the tangent plane at $f(q)$ is the vector space $tg(f,q) = \mathcal{T}_{f(q)} f(U) = D_q \; f(\mathbb{R}^d)$. Since T_f is a $\mathcal{D}$-immersion, for $\varphi \in D_*(U)$ and for ϵ small enough, $f_\epsilon = T_f * \varphi_\epsilon$ is a smooth immersion. We can thus define $tg(f_\epsilon, q) = (D_q \; f_\epsilon)(\mathbb{R}^d)$. In the smooth case, we would expect $tg(f_\epsilon, q)$ to tend to $tg(f,q)$ (see Figure 5.3):

Proposition 5.3. *The derivative of* $f_\epsilon = T_f * \varphi_\epsilon$ *is :*

$$D_q f_\epsilon = D_q(f * \varphi_\epsilon) = (D_q f * \varphi_\epsilon).$$

Moreover: $\lim_{\epsilon \to 0} D_q f_\epsilon = D_q f$.

Proof. This is a direct consequence of Lemma 3.2 and Theorem 3.5. □

5.4.2
Singular case: classical approach

A simple definition for tangent cones on continuous objects can be stated as:

Definition 5.4. *Given a set $K \in \mathbb{R}^n$, we say that $w \in \mathbb{R}^n$ belongs to the tangent cone at s to K, denoted by $T(s, K)$, if there exists a sequence $(h_m)_m \in (\mathbb{R}^n)^{\mathbb{N}}$ where $h_m \neq 0$ and a sequence $(\lambda_m)_m \in (\mathbb{R})^{\mathbb{N}}$ where $\lambda_m > 0$, such that:*

$$\begin{cases} h_m \xrightarrow[m\to\infty]{} w \\ \lambda_m \xrightarrow[m\to\infty]{} 0 \end{cases} \quad \textit{and} \quad \forall m, s + \lambda_m h_m \in M.$$

5.4.3
$\mathcal{D}$-tangent cone

Now, we intend to define a tangent cone directly from a $\mathcal{D}$-immersion. Following the regular case, the tangent cone of a $\mathcal{D}$-immersion T from parameter $q \in U$ would be the limit of $D_q T_\epsilon(\mathbb{R}^d)$, where $T_\epsilon = T * \varphi_\epsilon$ is a smooth immersion for small ϵ. This brute idea must overcome three delicate points: First, it would be a vector space convergence, and the tangent cone may not be a vector space. To overcome this, we can look at the function limit of $D_q T_\epsilon$. Second, for a general distribution, this may not converge to a function. We will thus look at the accumulation points instead of the limit. Last, this definition may depend of the approximation of the identity φ_ϵ used. Therefore we consider the union of the limits for all the approximations of the identity. This leads to the following definition:

Definition 5.5. *($\mathcal{D}$-Tangent Cone) Let T be a $\mathcal{D}$-immersion, $q \in U$ a fixed parameter. The $\mathcal{D}$-tangent cone of T at q denoted by $tg_{\mathcal{D}}(T, q)$, by:*

$$tg_{\mathcal{D}}(T, q) = \bigcup_{\varphi_\epsilon} \Big(Acc\,\{D_q\; T_\epsilon\} \Big)(\mathbb{R}^d),$$

where Acc denote the set of accumulation points in the L^1 topology.

The $\mathcal{D}$-tangent cone can be restricted by applying conditions on φ_ϵ.

Definition 5.6. *($\mathcal{D}^+$-Tangent Cone) Let T be a $\mathcal{D}$-immersion, $q \in U$ a fixed parameter. The restricted $\mathcal{D}$-tangent cone of T at q denoted by $tg^+_{\mathcal{D}}(T, q)$, by:*

$$tg^+_{\mathcal{D}}(T, q) = \bigcup_{\varphi_\epsilon > 0} \Big(Acc\,\{D_q\; T_\epsilon\} \Big)(\mathbb{R}^d) \qquad .$$

Similarly to this *positive tangent cone*, we can define the *symmetric tangent cone* by restricting the test functions φ_ϵ to be symmetric with respect to the origin.

Remark 5.7. *The $\mathcal{D}$-tangent cone is invariant by $\mathcal{D}$-change of parameters.*

Remark 5.8. *Proposition 5.3 ensures that, if f is an immersion, $tg_{\mathcal{D}}(T_f, q) = tg^+_{\mathcal{D}}(T_f, q) = tg(f, q)$, i.e. the $\mathcal{D}$-tangent cone extends the classical tangent cone.*

5.5 Intuitive proposal for $\mathcal{D}$-submanifold

The next step would be to combine $\mathcal{D}$-immersions in atlases to form $\mathcal{D}$-submanifolds, and to give an intrinsic definition for these objects. This section proposes a description of such objects in an informal way. We could define a $\mathcal{D}$-submanifolds $\mathcal{M}$ as a subset of $\mathbb{R}^n$ which is locally the limit of the images of $\mathcal{D}$-immersions:

$\mathcal{M} \subset \mathbb{R}^n$ is a $\mathcal{D}$-submanifold of dimension d if, for all point $x \in \mathcal{M}$, there exists :

- an open neighborhood V of x in $\mathbb{R}^n$,
- a compact K around V $(x \in V \subset K \subset \mathbb{R}^n)$,
- an open set U in $\mathbb{R}^d$,
- a $\mathcal{D}$-immersion T.

such that $\forall \varphi \in D_*(U)$, $\quad \big((T * \varphi_\epsilon)(U)\big) \cap K \xrightarrow[\epsilon \to 0]{d_H} \mathcal{M} \cap K$, where $d_H(A, B) = \max\{\sup_{a \in A}(d(a, B), \sup_{b \in B}(d(b, A)\}$ is the Hausdorff distance.
Moreover, if T and S satisfy the above criteria, then they must be compatible $\mathcal{D}$-immersions. This definition may be less restrictive if imposing only the existence of $\varphi \in D_*(U)$, instead of having the condition on all test function. The main challenge for this definition is to prove that a smooth submanifold is a $\mathcal{D}$-submanifold. This may be easier with the convergence in K, as suggested above, since the Hausdorff distance is reached on compacts.

6
Examples

We experiment our constructions on three submanifolds immersed in the plane: a parabola as an example of a smooth submanifold and a V-function and a cusp for non-smooth submanifolds.

6.1
Test functions

To experiment the concept of $\mathcal{D}$-immersions, we need some test functions in order to evaluate our maps.

To do so, we use combinations of the following standard test function:

$$\phi : t \longmapsto \begin{cases} \frac{\exp(t^2/(t^2-1))}{NORM} & if\ t^2 < 1 \quad , \\ 0 & otherwise \quad . \end{cases}$$

From this test function it is possible to generate many others using a very simple trick: averaging several copies of ϕ translated to different points on the

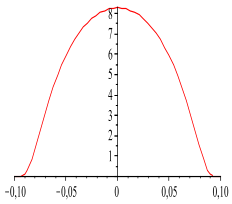
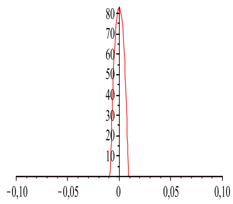
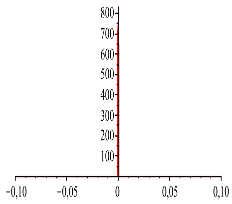
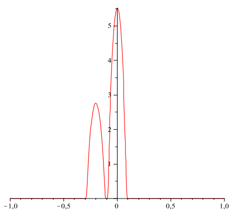
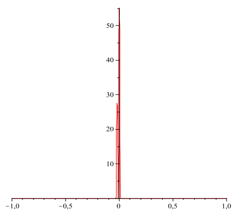
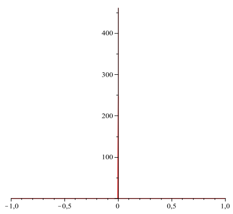
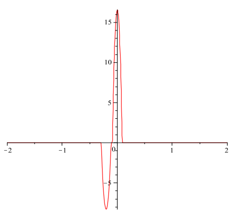
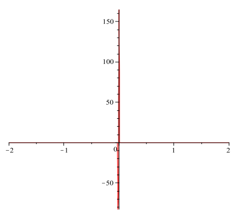
φ	$\epsilon = 0.1$	$\epsilon = 0.01$	$\epsilon = 0.001$
$\phi(t)$			
$\frac{2}{3}\phi(t) + \frac{1}{3}\phi(t+2)$			
$2\phi(t) - \phi(t+2)$			

Table 6.1: Some of the approximations of the identity used in the examples.

real line we get a new test function (see Table 6.1). We can easily then create an approximation of the identity by:

$$\varphi_\epsilon : t \longmapsto \frac{1}{\epsilon^d} \cdot \frac{\sum \alpha_i \cdot \phi(\frac{t-t_i}{\epsilon})}{\sum \alpha_i}.$$

Through our work we used different test functions to examplify $\mathcal{D}$-immersions upon two criterias: their positiveness and their symmetry with respect to the origin. In paricular, we expect positive test functions to respect convexity properties, in particular for the positive tangent cone, and symmetric test functions to respect the symmetries of the immersion. That leads us to consider five types of test functions: a basic gaussian-like test function (φ^1), a positive and symmetric one (φ^2), a non-positive and symmetric (φ^3), a positive and non-symmetric one (φ^4) and finally a non-positive and non-symmetric one (φ^5). Here is their expression in function of ϕ:

$$\begin{aligned}
\varphi^1 &= \phi(t),\\
\varphi^2 &= \frac{\phi(t-1)+\phi(t+1)}{2},\\
\varphi^3 &= \frac{2\phi(t-1)+\phi(t+1)}{3},\\
\varphi^4 &= -\phi(t-1)+3\phi(t)-\phi(t+1),\\
\varphi^5 &= 2\phi(t-1)-\phi(t+1).
\end{aligned}$$

6.2 Experimental setup

In order to estimate the direction of the tangent plane and the curvature at a given point of an approximated immersion in the plane we use a Maple-based program (see appendix A). We use the five test functions $\varphi^1 \dots \varphi^5$ described previously to proceed with the tests. Three parameters are to be set to compute a test. The main one is the value of ϵ, the parameter relative to the approximation of the identity. We fixed three values for ϵ along our tests: 0.1, 0.05 and 0.01. Another parameter named translation sets the overlapping between the different bumps a test function can have. We set it to $\frac{9}{10}$ in order to have a small overlapping between the bumps. As a matter of fact, we observed that a small overlap allows a cleaner convolution between the test function and the immersion. The last parameter to set is the number of digits we want Maple to work with for numerical evaluations, although Maple tries to perform most of the evaluations formally. A greater number of digits oftenly leads to computational issues and we thus try to optimize the value of this parameter

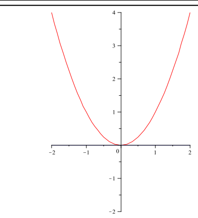
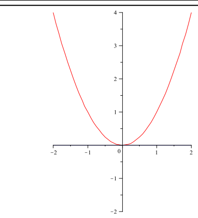
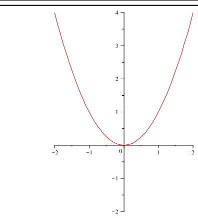
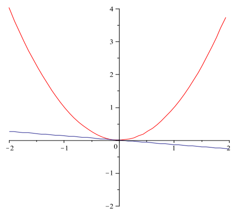
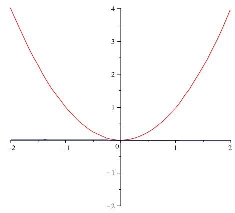
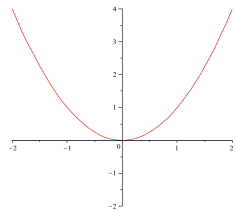
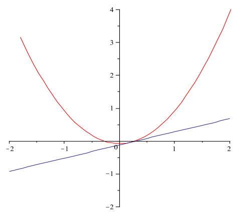
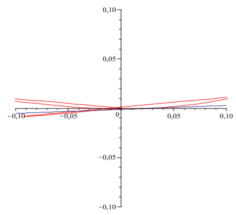
M_φ	$\epsilon = 0.1$	$\epsilon = 0.01$	$\epsilon = 0.001$
$\phi(t)$			
$\frac{2}{3}\phi(t) + \frac{1}{3}\phi(t+2)$			
$2\phi(t) - \phi(t+2)$			

Table 6.2: Mean manifolds with the mean tangent line for the parabola.

for each test. We fix it according to the value of the ϵ parameter: 5, 8 and 10 digits for $\epsilon = 0.1, 0.05, 0.01$ respectively.

Finally, we choose a parabola, a V-function and a cusp to be our experimental immersions. All three of them are $\mathcal{D}$-immersions since they are all graphs of functions. Both non-smooth immersions are representative of low order singularities in classical geometry, and are thus two interesting samples in discrete geometry. In particular, the V-function is the typical example of polygonal curves which have many applications in discrete modelling.

We try the three dimensional case (see appendix B) as well, meanwhile we encountered several issues when executing the Maple data-sheet. Although we managed to produce correct three dimensional test functions from the base function ϕ, we could not have Maple compute numerical values of the convolution. This is due to the lack of numerical methods for double integrals in Maple. Actually, Maple computes the double integral of the convolution by iterated integrals, requiring a formal integration followed by a numerical integration. Since ϕ has no simple primitive, Maple is not able to perform the formal integration and thus cannot evaluate or plot the required convolutions.

6.3 Tests on a parabola

The first example is a C^∞ submanifold of dimension 1 immersed in the plane; namely a parabola parameterized as the graph of the square function

Test function	$\epsilon = 0.1$ Tang	Curv	$\epsilon = 0.05$ Tang	Curv	$\epsilon = 0.01$ Tang	Curv
φ^1	0	2	0	2	0	2
φ^2	0	2	0	2	0	2
φ^3	-0.06	1.98	-0.03	1.99	-0.006	1.99
φ^4	0	2	0	1.99	0	1.99
φ^5	-0.54	1.36	-0.27	1.79	-0.054	1.99

Table 6.3: Tangent plane direction and curvature estimation for the parabola.

(see Table 6.2). Define:

$$M = \Big\{(t, t^2), t \in \mathbb{R}\Big\}.$$

Let U be an open of $\mathbb{R}$, define f such that:

$$f : \begin{cases} U & \longrightarrow M \\ t & \longmapsto (t, t^2) \end{cases} ,\ .$$

The function f is an immersion on M and as being the graph of a function theorem 4.3 ensures that T_f is a $\mathcal{D}$-immersion. Now let's see how this is related to the classical theory. We define f_ϵ as before:

$$f_\epsilon : \begin{cases} U & \longrightarrow M_{\varphi_\epsilon} \\ x & \longmapsto (f * \varphi_\epsilon)(x) \end{cases} ,\ .$$

For all φ_ϵ such that f_ϵ is an immersion, M_{φ_ϵ} is a mean submanifold of the plane. We can observe on Table 6.2 nice approximations for positive test functions, while non-positive test functions may generate some instabilities. Table 6.3 gives the estimations of the direction of the tangent plane $Tang$ and the curvature $Curv$ at parameter value 0 of the immersion, based on the five test functions listed previously.

$$Tang = \frac{y'(0)}{x'(0)} \qquad Curv = \frac{x'(0)y''(0) \ - \ x''(0)y'(0)}{(x'(0)^2 \ + \ y'(0)^2\ )^{\frac{3}{2}}}$$

Notice that for all symmetric test functions (i.e, φ^1, φ^2, φ^4) the tangent plane and the curvature are well approximated. For non-symmetric test functions, both the tangent plane and the curvature converge to their original values when ϵ tends to 0.

6.4 Tests on a V-function

To test our theory on topological submanifolds, we study the graph of the absolute value function. We are interested in studying the unique singularity

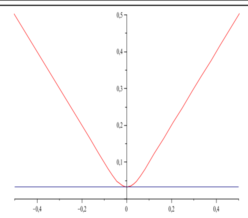
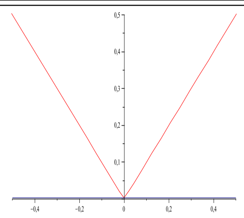
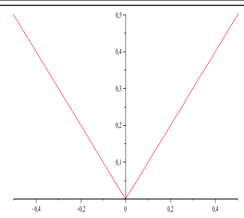
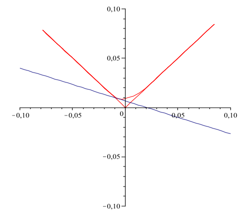
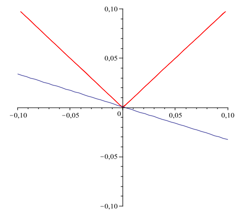
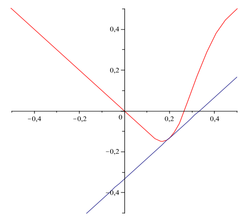
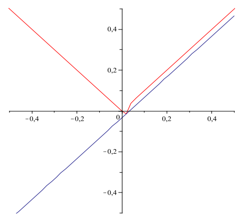
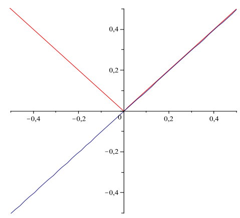
M_φ	$\epsilon = 0.1$	$\epsilon = 0.01$	$\epsilon = 0.001$
$\phi(t)$			
$\frac{2}{3}\phi(t) + \frac{1}{3}\phi(t+2)$			
$2\phi(t) - \phi(t+2)$			

Table 6.4: Mean manifolds with the mean tangent line for the V-shape.

of this submanifold which stands at the origin of the plane (see Table 6.4). Define:

$$N = \Big\{(t, |t|), t \in \mathbb{R}\Big\}.$$

Let V be an open of $\mathbb{R}$, define g such that:

$$g : \begin{cases} V & \longrightarrow N \quad , \\ t & \longmapsto (t, |t|) \quad . \end{cases}$$

The function g is a parameterization of N but it fails to be an immersion at $t = 0$. However by Theorem 4.3 T_g is a (non-trivial) $\mathcal{D}$-immersion. Now define g_ϵ as being the mean map of g:

$$g_\epsilon : \begin{cases} V & \longrightarrow N_{\varphi_\epsilon} \quad , \\ x & \longmapsto (g * \varphi_\epsilon)(x) \quad . \end{cases}$$

Therefore N_{φ_ϵ} is a mean submanifold of the plane for all $\epsilon > 0$, i.e, a tangent plane can be defined at any point. We can observe on Table 6.4 that non-symmetric test functions generate non horizontal tangent planes, even when ϵ is reduced. While symmetric test functions respect the symmetry of the right angle. We can actually prove this fact: Name s the unique singularity of N.

When ϵ tends to 0, we can compute the $\mathcal{D}$-tangent cone of T_g at s:

$$tg_{\mathcal{D}}(T_g, s) = \bigcup_{\varphi_\epsilon > 0} \Big(Acc\,\{D_s\ g_\epsilon\} \Big)(\mathbb{R}) \qquad .$$

Now looking at the $\mathcal{D}^+$-tangent cone of T_g at s, we have:

Proposition 6.1. *$tg^+_{\mathcal{D}}(T_g, s)$ respects the convexity of the submanifold N: all its elements are directions below N at s.*

Proof. Since N is a graph in the plane, it can parameterized by two functions, $x(t) = t$ and $y(t) = |t|$. Here $tg^+_{\mathcal{D}}(T_g, s)$ is below N when: $-1 \leq \dfrac{y'(0)}{x'(0)} \leq 1$. Theorem 4.3 ensures that $x'(0) = 1$:

$$x'(0) = -\int_{-\infty}^{\infty} t \cdot \varphi'_\epsilon(t)\ dt = -\Big[t \cdot \varphi_\epsilon(t)\Big]_{-\infty}^{\infty} + \int_{-\infty}^{\infty} \varphi_\epsilon(t)\ dt = 1.$$

Now, the derivative of y at 0 is:

$$\begin{aligned} y'(0) &= -\int_{-\infty}^{\infty} |t| \cdot \varphi'_\epsilon(t)\ dt \\ &= -\int_{-\infty}^{0} (-t) \cdot \varphi'_\epsilon(t)\ dt - \int_{0}^{\infty} t \cdot \varphi'_\epsilon(t)\ dt \\ &= \Big[t \cdot \varphi_\epsilon(t)\Big]_{-\infty}^{0} - \int_{-\infty}^{0} \varphi_\epsilon(t)\ dt - \Big[t \cdot \varphi_\epsilon(t)\Big]_{0}^{\infty} + \int_{0}^{\infty} \varphi_\epsilon(t)\ dt \\ &= \int_{0}^{\infty} \varphi_\epsilon(t)\ dt - \int_{-\infty}^{0} \varphi_\epsilon(t)\ dt + \int_{-\infty}^{0} \varphi_\epsilon(t) - \int_{-\infty}^{0} \varphi_\epsilon(t) \\ &= x'(0) - 2 \cdot \int_{-\infty}^{0} \varphi_\epsilon(t) \end{aligned}$$

We obtain:

$$\frac{y'(0)}{x'(0)} = 1 - 2 \cdot \int_{-\infty}^{0} \varphi_\epsilon(t).$$

Since $\int_{-\infty}^{\infty} \varphi_\epsilon(t)\ dt = 1$, $\int_{-\infty}^{0} \varphi_\epsilon(t)\ dt \leq 1$. Moreover, since $\varphi_\epsilon \geq 0$, $1 \geq \int_{-\infty}^{0} \varphi_\epsilon(t)\ dt > 0$, we get:

$$-1 \leq 1 - 2 \cdot \int_{-\infty}^{0} \varphi_\epsilon(t) < 1.$$

And finally

$$-1 \leq \frac{y'(0)}{x'(0)} \leq 1.$$

□

Test function	$\epsilon = 0.1$		$\epsilon = 0.05$		$\epsilon = 0.01$	
	Tang	**Curv**	**Tang**	**Curv**	**Tang**	**Curv**
φ^1	0	16.57	0	33.14	0	165.7
φ^2	0	0.23	0	0.46	0	2.33
φ^3	-0.33	0.19	-0.33	0.39	-0.33	1.99
φ^4	0	49.24	0	98.48	0	492.4
φ^5	-2.99	0.007	-2.99	0.014	-2.99	0.073

Table 6.5: Tangent plane direction and curvature estimation for the V-function.

Similarly to the previous example, Table 6.5 gives the estimations of the direction of the tangent plane and the curvature at parameter value 0 of the immersion. Here it can be observed that the curvature is inversely proportional to ϵ: when ϵ is divided by a certain amount, the curvature is multiplied by the same amount. This was an expected result since when ϵ tends to 0 the mean submanifold approximates the right angle with increasing precision linearly and thus the curvature rises linearly. Notice that here again symmetric test functions generate good approximations of the tangent plane.

6.5 Tests on a cusp

For the cusp, we only computed the estimations for the direction of the tangent plane and the curvature at parameter value 0 of the immersion (see Table 6.6). Once again due to the symmetry of the immersion, the direction of the tangent plane is well approximated when using symmetric test functions. The curvature explodes in absolute value when ϵ tends to 0: this behaviour corresponds to the non-linear structure of the cusp. Since it is a highly singular curve at its origin, approximating with convolution with a low valued ϵ results in a bad approximation of the curvature. When ϵ decreases the curvature rises rapidly. We can observe a lack of information around the point we focused on when testing with φ^3 since it has no symmetry. Moreover for test function φ^5, the weight on negative parts lead to a very low convergence and high numerical instability for the tangent plane, and flat approximations for the curvature. We

Test function	$\epsilon = 0.1$		$\epsilon = 0.05$		$\epsilon = 0.01$	
	Tang	**Curv**	**Tang**	**Curv**	**Tang**	**Curv**
φ^1	0	36.29	0	102.6	0	1147.8
φ^2	0	-18.06	0	-51.09	0	-571.3
φ^3	-0.62	-11.04	-0.88	-21.57	-1.97	-52.95
φ^4	0	145.03	0	410.2	0	4586.1
φ^5	-5.60	-0.097	-7.93	-0.1	-17.7	-0.101

Table 6.6: Tangent plane direction and curvature estimation for the cusp.

would expect a similar result as property 6.1 for the positive tangent cone of the cusp, although in that case the tangent cone should be reduced to a single direction. Finally, it is trivial to see that the symmetric tangent cone for this immersion is actually reduced to the vertical direction.

7
Conclusion

Along this dissertation we developed a formulation through which we extend geometric quantities to non-smooth objects. In particular we address the delicate task to define a tangent object at a singularity. Although there exists many ways to approximate such tangent spaces, most of them are rather complex and intricate. Starting from classical geometric theory where submanifolds are parameterized through differential applications, we propose to use distributions in order to differentiate raw continuous parameterizations. Although there exists a great difference between differential and singular geometry, our main objective was to build up a direct geometric formulation to reduce that difference.

The $\mathcal{D}$-geometric notions were designed via distribution theory precisely for that purpose. As distributions generalize functions, we used them to substitute classical parameterizations in order to obtain a new type of parameterizations. Building up on the concept of immersion, we managed to generalize some fundamentals tools of differential geometry. $\mathcal{D}$-immersions generalize classical immersions, in the sense that the distribution associated to an immersion is a $\mathcal{D}$-immersion. Moreover they strictly extend immersions as we exhibit non-trivial $\mathcal{D}$-immersions such as graphs of L^1 functions.

$\mathcal{D}$-submanifolds would generalize smooth submanifolds in the sense that $\mathcal{D}$-immersions associated to parameterizations of a smooth submanifold M define the $\mathcal{D}$-submanifold associated to M. However we have not managed to extend classical submanifolds as we thought we could do. A great difficulty remains when trying to extend our results from C^1 to C^0 and define flexible notions for compatible $\mathcal{D}$-immersions. This happens here mainly because of the proper distribution theory which does not allow C^1 test functions. Meanwhile this work provided answers and lead us to new questions. We further experimented these notions applying them to tangent cone approximations and curvature estimations.

Other properties of $\mathcal{D}$-immersions might be discovered and we may hope that $\mathcal{D}$-submanifolds could someday extend classical differential submanifolds. We ended the fourth chapter by introducing a new notion of tangent cone, in our attempt to study singularities. It is beyond the scope of this dissertation to proceed to a deeper analysis of this definition but it is a stimulating problem for future works. In particular, we would like to further study the relations between different definitions of tangent cones especially in the light of applications to singularities analysis.

Bibliography

[Federer 1996] FEDERER, H.. **Geometric Measure Theory**. Springer, 1996.

[Langevin 2006] LANGEVIN, R.. **Integral geometry from Buffon to the use of twentieth century mathematics**, 1996.

[Lebeau 1999] LEBEAU, G.. **Théorie des distributions et analyse de Fourier**. École Polytechnique, 1999.

[Morgan 2000] MORGAN, F.. **Geometric Measure Theory: A Beginner's Guide**. Academic Press, London, UK, 2000.

[Santaló 1953] SANTALÓ, L. A.. **Introduction to integral geometry**. Université de Nancago, 1953.

[Schwartz 1997] SCHWARTZ, L.. **Un mathématicien aux prises avec le siècle**. Editions Odile Jacob, Paris, France, 1997.

[do Carmo 2005] DO CARMO, M. P.. **Geometria Diferencial de Curvas e Superfícies**. Sociedade Brasilieira de Matemática, Rio de Janeiro, Brazil, 2005.

A
Maple worksheet for the 2d case

A.0.1
Global parameters

Accuracy

Digits := 5 :

Epsilon

epsilon := 0.1 :

Overlapping

trans := 9/10 :

Graphics viewport

xm := -1.1 - trans : xM := +1.1 + trans : ym := -1.1 : yM := +3.1 :

Immersion

X := t -> t ; Y := t -> sqrt(abs(t)) ;

$$X := t \mapsto t$$
$$Y := t \mapsto \sqrt{|t|}$$

A.0.2
Test functions

```
> phi_unnorm := t-> piecewise( -1<t and t<1, exp((-t^2)/(1-t^2)),
0 );
```

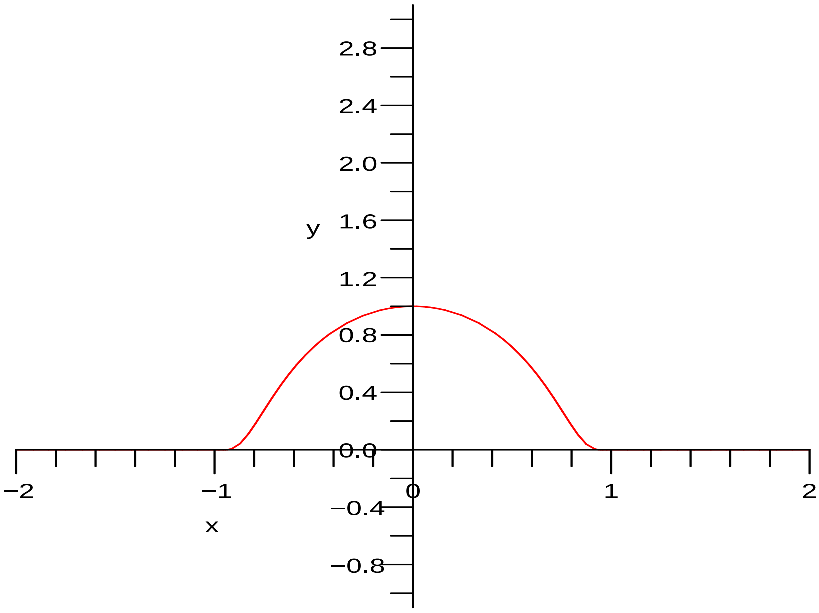
$$phi\_unnorm := (s,t) \mapsto \begin{cases} e^{-\frac{s^2+t^2}{1-s^2-t^2}} & s^2+t^2 < 1 \\ 0 & otherwise \end{cases}$$

```
> Digits_cpy := Digits :
> Digits := max( Digits, 20 ) :
> norm_phi := evalf( Int( phi_unnorm(t), t=-1..1 ) ) ;
> Digits := Digits_cpy :
```

$$norm\_phi := 1.2069003224378761753$$

```
> phi := t -> phi_unnorm(t) / norm_phi :
> plot( phi_unnorm(x), x=xm..xM, y=ym..yM, scaling=constrained )
;
```

$$\phi := t \mapsto \frac{phi\_unnorm(t)}{norm\_phi}$$




```
> TFonction := proc( a,m )
>   local A, f, fe, dfe, ddfe ;
>   global phi, epsilon, trans ;
>
>   A := sum( a[i], i=1..nops(a) ) ;
>
>   f    := simplify( t -> sum( a[i]*phi(t+m[i]*trans),
i=1..nops(a) ) / A ) ;
>   fe   := t -> f(t/epsilon) / epsilon ;
>   dfe  := simplify( t -> subs( u=t, diff(  fe(u),u ) ) ) ;
>   ddfe := simplify( t -> subs( u=t, diff( dfe(u),u ) ) ) ;
>
>   print( plot( [ epsilon * fe( epsilon * x ), epsilon^2 *
dfe( epsilon * x ), epsilon^3 * ddfe( epsilon * x ) ], x=xm..xM,
y=ym..yM, scaling=constrained, color=[red, navy, green] ) ) ;
>   return fe, dfe, ddfe ;
> end proc :
```

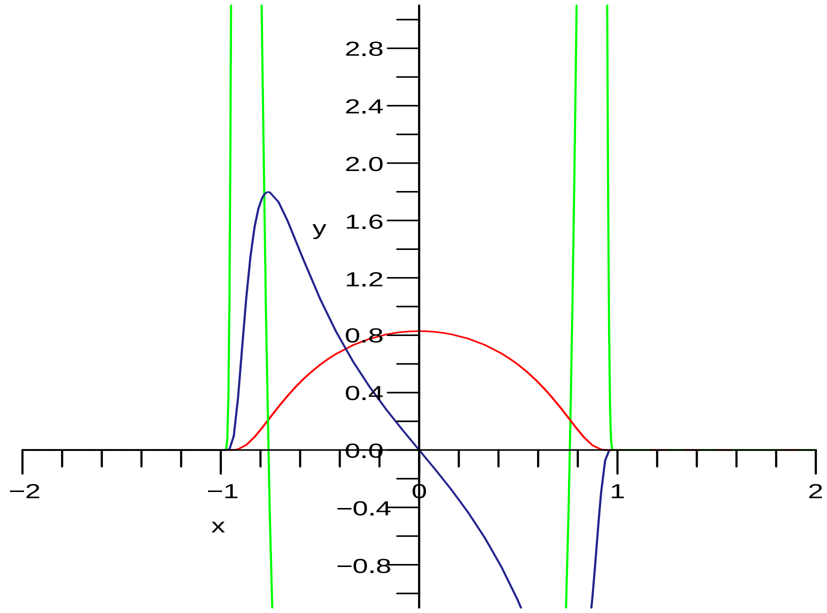
A.0.3 Convolutions

```
> convole := proc( phi, dphi, ddphi, q )
>   global X,Y, trans, xm,xM,ym,yM, epsilon ;
>   local x,y, dx,dy, ddx, ddy, x0,y0, dx0,dy0, ddx0, ddy0, tg,
ntg, crv, tm,tM ;
>
>   tm   := epsilon * (-1.0-trans) ;
>   tM   := epsilon * ( 1.0+trans) ;
>
>   x    := z -> evalf( Int( X(z-t) * phi  (t), t=tm..tM ) ) ;
>   y    := z -> evalf( Int( Y(z-t) * phi  (t), t=tm..tM ) ) ;
>   dx   := z -> evalf( Int( X(z-t) * dphi (t), t=tm..tM ) ) ;
>   dy   := z -> evalf( Int( Y(z-t) * dphi (t), t=tm..tM ) ) ;
>   ddx  := z -> evalf( Int( X(z-t) * ddphi(t), t=tm..tM ) ) ;
>   ddy  := z -> evalf( Int( Y(z-t) * ddphi(t), t=tm..tM ) ) ;
>
>   x0   := x  (q) ;
>   y0   := y  (q) ;
>   dx0  := dx (q) ;
>   dy0  := dy (q) ;
>   ddx0 := ddx(q) ;
>   ddy0 := ddy(q) ;
>
>   tg   := evalf( dy0 / dx0 ) ;
>   ntg  := sqrt( dx0^2 + dy0^2 ) ;
>   crv  := evalf( ( dx0*ddy0 - dy0*ddx0 ) / ntg^3 ) ;
>
>   print( "At ", x0,y0, " tg=", tg, " crv=", crv ) ;
>
>   plot( [ [x(z),y(z),z=-1..1], tg*(z-x0) + y0, [x0-dy0/crv/ntg
+ sin(theta)/crv,y0+dx0/crv/ntg - cos(theta)/crv,theta=0..2*Pi]
], x=epsilon*xm..epsilon*xM, y=epsilon*ym..epsilon*yM, scaling=constrained,
color=[red,navy,green], thickness=[3,5,2] ) ;
>  end proc :
```

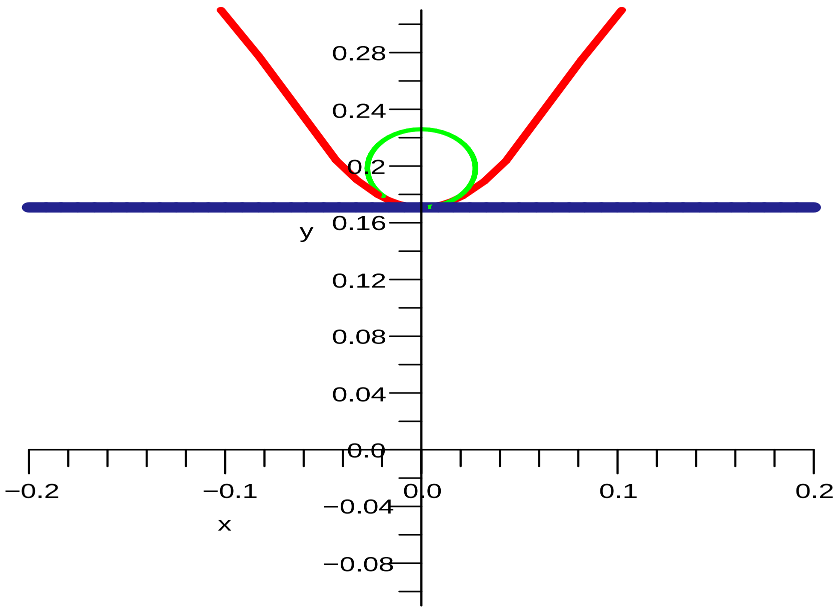
A.0.4
Results

Basic test function

phie, dphie, ddphie := TFonction([3],[0]) : convole(phie, dphie, ddphie, 0) ;



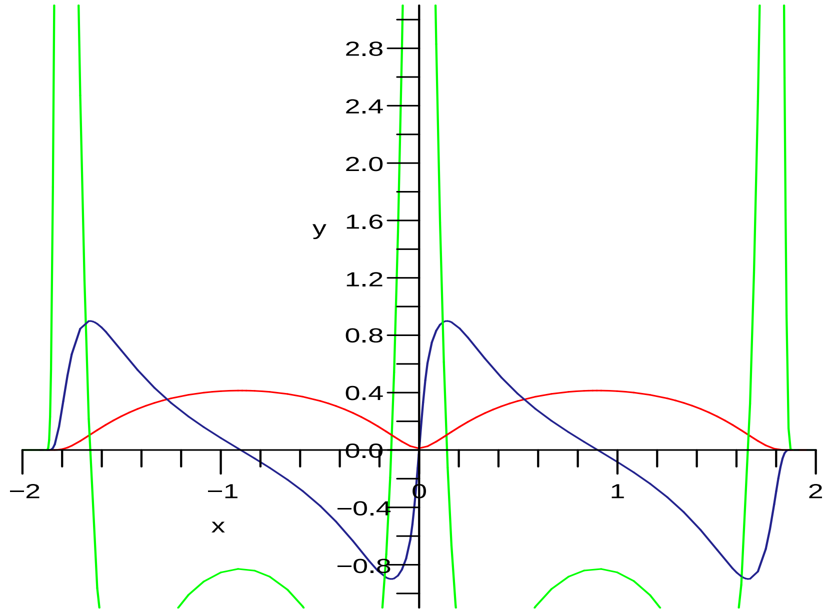
"At", 0.0, 0.17085, "$tg =$", 0.0, "$crv =$", 36.299



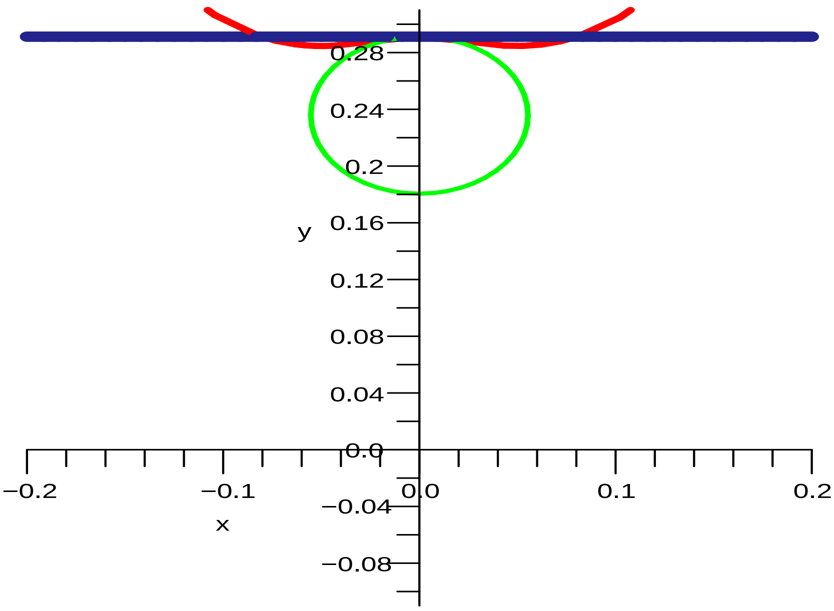
Positive and symmetric double bump test function

phie_2sym, dphie_2sym, ddphie_2sym := TFonction([1,1], [-1,+1])

: convole(phie_2sym, dphie_2sym, ddphie_2sym, 0) ;

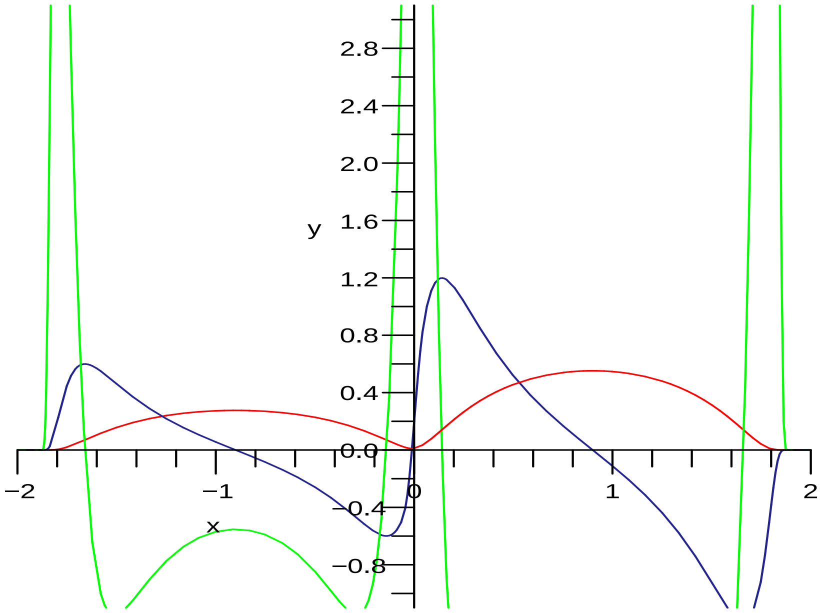


“*At*”, 0.0, 0.29122, “*tg* =”, 0.0, “*crv* =”, −18.065



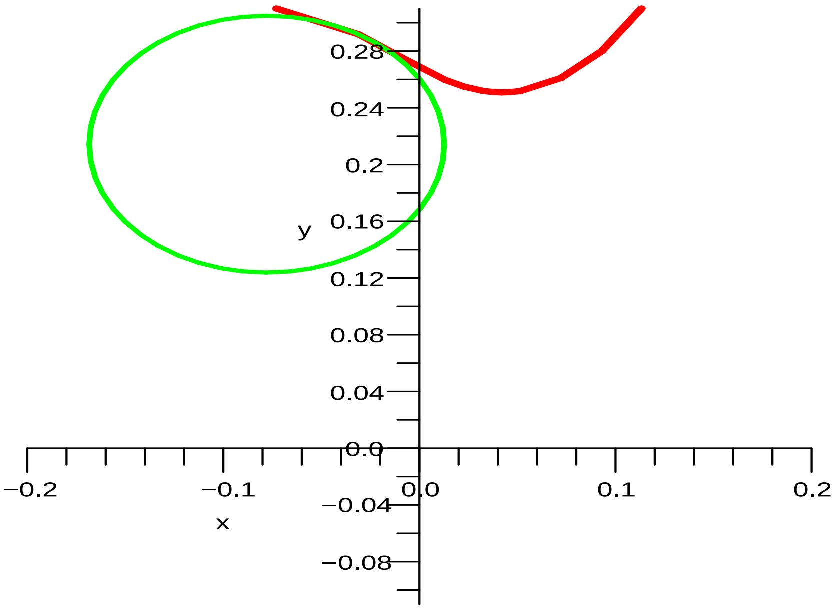
Positive and non-symmetric double bump test function

phie_2dis, dphie_2dis, ddphie_2dis := TFonction([2,1], [-1,+1]) :
convole(phie_2dis, dphie_2dis, ddphie_2dis, 0) ;



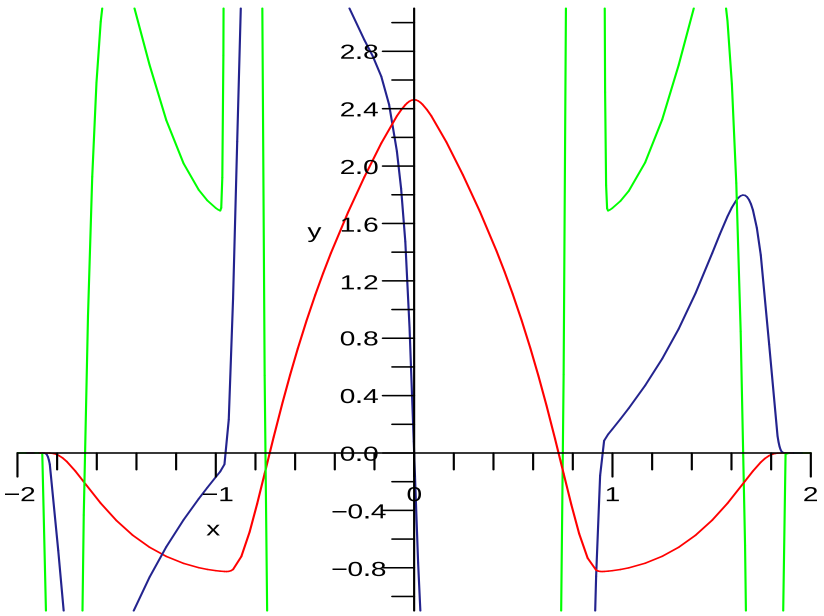
"At", −0.029999, 0.29122, "$tg =$", −0.62308, "$crv =$", −11.045

```
Warning, unable to evaluate 1 of the 3 functions to numeric
values in the region; see the plotting command's help page to
ensure the calling sequence is correct
```

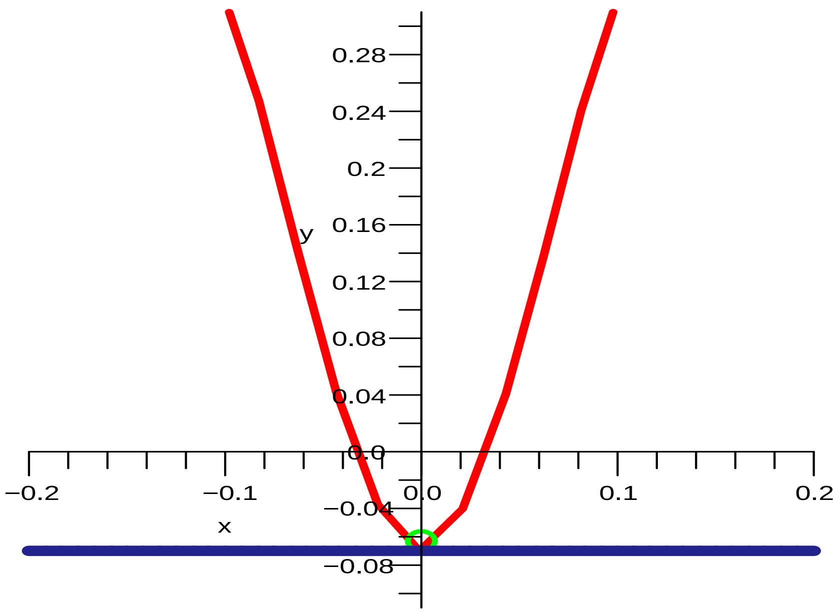


Non-positive and symmetric triple bump test function

phie_nsym, dphie_nsym, ddphie_nsym := TFonction([-1,3,-1], [-1,0,+1]) : convole(phie_nsym, dphie_nsym, ddphie_nsym, 0) ;



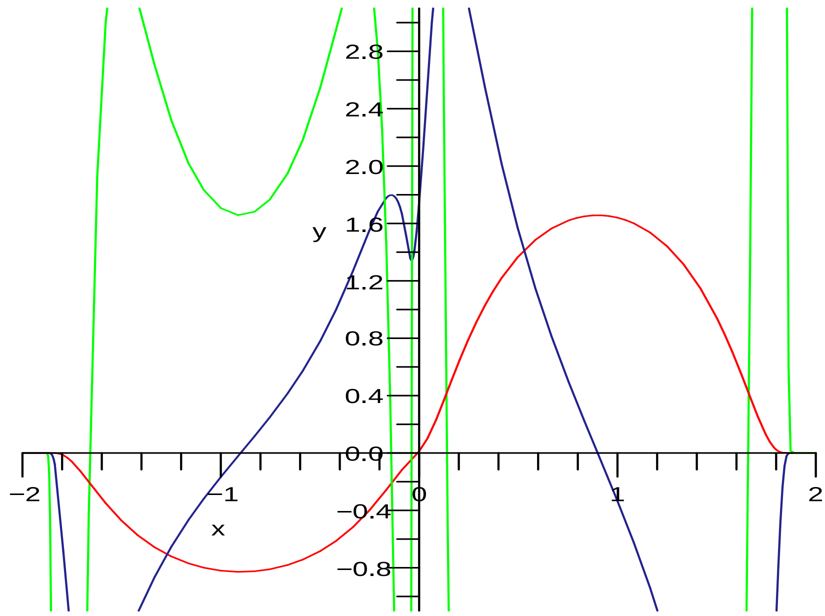
"*At*", 0.0, −0.069909, "*tg* =", 0.0, "*crv* =", 145.03



Non-positive and non-symmetric double bump test function

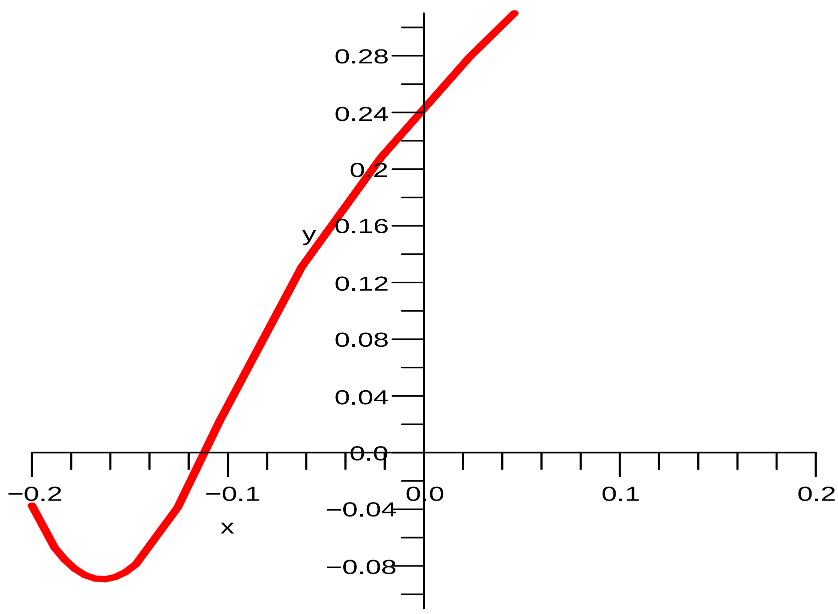
phie_ndis, dphie_ndis, ddphie_ndis := TFonction([2,-1], [-1,+1]) :

convole(phie_ndis, dphie_ndis, ddphie_ndis, 0) ;



$$\text{“}At\text{”},\ -0.27000,\ 0.29121,\ \text{“}tg =\text{”},\ -5.6081,\ \text{“}crv =\text{”},\ -0.097733$$

```
Warning, unable to evaluate 1 of the 3 functions to numeric
values in the region; see the plotting command's help page to
ensure the calling sequence is correct
```



B
Maple worksheet for the 3d case

B.0.5
Global parameters

Accuracy

Digits := 10 :

Epsilon

epsilon := 0.1 :

Overlapping

trans := 9/10 :

Graphics viewport

xm := -1.1 - trans : xM := +1.1 + trans : ym := -1.1 - trans : yM := +1.1 + trans : zm := -1.1 : zM := +3.1 :

Immersion

X := (s,t) -> s ; Y := (s,t) -> t ; Z := (s,t) -> s^2 + t^2 ;

$$X := (s,t) \mapsto s$$

$$Y := (s,t) \mapsto t$$

$$Z := (s,t) \mapsto s^2 + t^2$$

B.0.6
Test functions

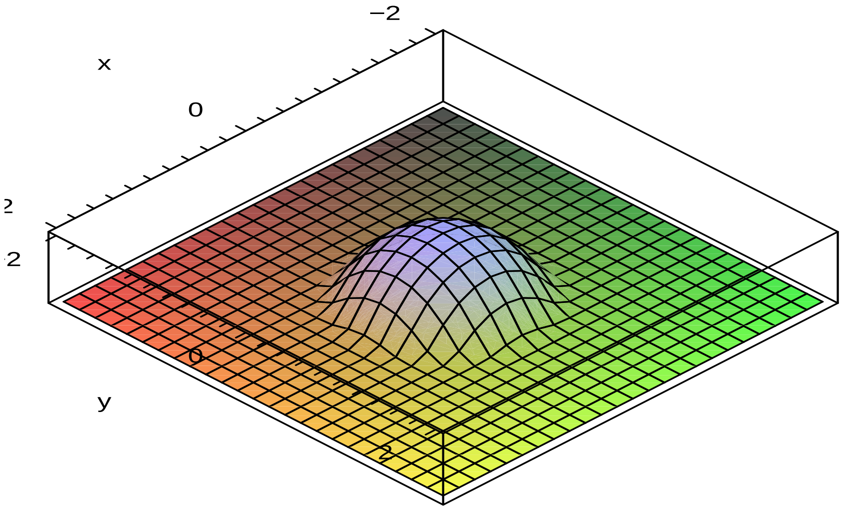
```
> phi_unnorm := (s,t)-> piecewise( (s^2) + (t^2) <1, exp(-(s^2 +
t^2)/(1-s^2 - t^2)), 0 );
```

$$phi\_unnorm := (s,t) \mapsto \begin{cases} e^{-\frac{s^2+t^2}{1-s^2-t^2}} & s^2+t^2 < 1 \\ 0 & otherwise \end{cases}$$

```
> Digits_cpy := Digits :
> Digits := max( Digits, 10 ) :
> norm_phi := evalf( Int(Int( phi_unnorm(s,t), s=-1..1), t=-1..1
) ) ;
> Digits := Digits_cpy :
```

$$norm\_phi := 1.268112161$$

```
> phi := (s,t) -> phi_unnorm(s,t) / norm_phi :
> plot3d( phi_unnorm(x,y), x=xm..xM, y=ym..yM, scaling=constrained,
axes=boxed ) ;
```




```
> TFonction := proc( a,m,n )
>   local A, f, fe, dfeu, dfev, ddfeu, ddfev, ddfeuv ;
>   global phi, epsilon, trans ;
>
>   A := sum( a[i], i=1..nops(a) ) ;
>
>   f    := simplify( (s,t) -> sum( a[i]*phi(s+m[i]*trans,
t+n[i]*trans), i=1..nops(a) ) / A ) ;
>   fe   := (s,t) -> f(t/epsilon, s/epsilon) / epsilon ;
>   dfeu  := simplify( (s,t) -> subs( u=s, v=t, diff(  fe(u,v),u
) ) ) ;
>   dfev  := simplify( (s,t) -> subs( u=s, v=t, diff(  fe(u,v),v
) ) ) ;
>   ddfeu := simplify( (s,t) -> subs( u=s, v=t, diff( fe(u,v),u,u
) ) ) ;
>   ddfev := simplify( (s,t) -> subs( u=s, v=t, diff( fe(u,v),v,v
) ) ) ;
>   ddfeuv := simplify( (s,t) -> subs( u=s, v=t, diff( fe(u,v),u,v
) ) ) ;
>
>   print( plot3d( [ epsilon * fe ( epsilon * x, epsilon * y )
], x=-2..2, y=-2..2, scaling=constrained ) ) ;
> return fe, dfeu, dfev, ddfeu, ddfev, ddfeuv ;
> end proc :
```

B.0.7 Convolutions

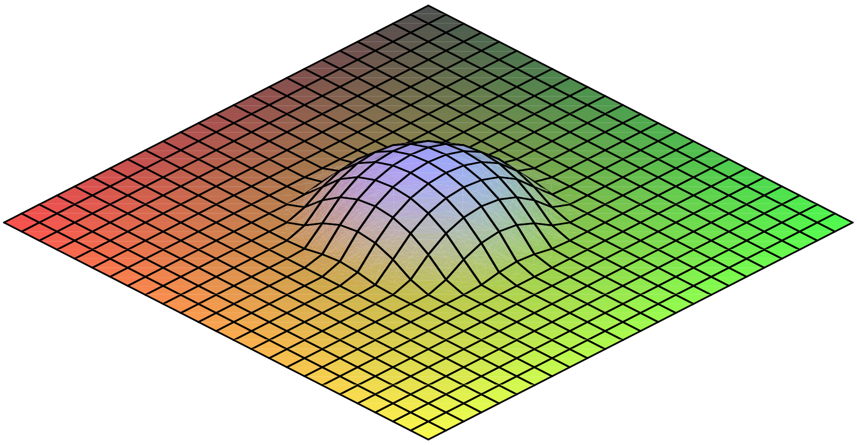
```
> convole := proc( phi, dphiu, dphiv, ddphiu, ddphiv, ddphiuv,
q, p )
>    global X,Y,Z, trans, xm,xM,ym,yM, epsilon ;
>    local x,y,z, dxu,dxv,dyu,dyv,dzu,dzv, ddxu,ddxv,ddxuv,ddyu,ddyv,ddyuv,ddzu,ddzv
x0,y0,z0, dxu0,dxv0,dyu0,dyv0,dzu0,dzv0, ddxu0,ddxv0,ddxuv0,ddyu0,ddyv0,ddyuv0,ddzu0
tg, tm, tM, crv ;
>
>    tm     := epsilon * (-1.0-trans) ;
>    tM     := epsilon * ( 1.0+trans) ;
>
>    x      := (u,v) -> evalf( int(int( X(u-s,v-t) * phi  (s,t),
s=tm..tM), t=tm..tM ) ) ;
>    y      := (u,v) -> evalf( int(int( Y(u-s,v-t) * phi  (s,t),
s=tm..tM), t=tm..tM ) ) ;
>    z      := (u,v) -> evalf( int(int( Z(u-s,v-t) * phi  (s,t),
s=tm..tM), t=tm..tM ) ) ;
>
>    dxu    := (u,v) -> evalf( int(int( X(u-s,v-t) * dphiu
(s,t), s=tm..tM), t=tm..tM ) ) ;
>    dxv    := (u,v) -> evalf( int(int( X(u-s,v-t) * dphiv
(s,t), s=tm..tM), t=tm..tM ) ) ;
>    dyu    := (u,v) -> evalf( int(int( Y(u-s,v-t) * dphiu
(s,t), s=tm..tM), t=tm..tM ) ) ;
>    dyv    := (u,v) -> evalf( int(int( Y(u-s,v-t) * dphiv
(s,t), s=tm..tM), t=tm..tM ) ) ;
>    dzu    := (u,v) -> evalf( int(int( Z(u-s,v-t) * dphiu
(s,t), s=tm..tM), t=tm..tM ) ) ;
>    dzv    := (u,v) -> evalf( int(int( Z(u-s,v-t) * dphiv
(s,t), s=tm..tM), t=tm..tM ) ) ;
>
>    ddxu   := (u,v) -> evalf( int(int( X(u-s,v-t) * ddphiu
(s,t), s=tm..tM), t=tm..tM ) ) ;
>    ddxv   := (u,v) -> evalf( int(int( X(u-s,v-t) * ddphiv
(s,t), s=tm..tM), t=tm..tM ) ) ;
>    ddxuv  := (u,v) -> evalf( int(int( X(u-s,v-t) * ddphiuv
(s,t), s=tm..tM), t=tm..tM ) ) ;
>    ddyu   := (u,v) -> evalf( int(int( Y(u-s,v-t) * ddphiu
(s,t), s=tm..tM), t=tm..tM ) ) ;
>    ddyv   := (u,v) -> evalf( int(int( Y(u-s,v-t) * ddphiv
(s,t), s=tm..tM), t=tm..tM ) ) ;
>    ddyuv  := (u,v) -> evalf( int(int( Y(u-s,v-t) * ddphiuv
(s,t), s=tm..tM), t=tm..tM ) ) ;
>    ddzu   := (u,v) -> evalf( int(int( Z(u-s,v-t) * ddphiu
(s,t), s=tm..tM), t=tm..tM ) ) ;
>    ddzv   := (u,v) -> evalf( int(int( Z(u-s,v-t) * ddphiv
(s,t), s=tm..tM), t=tm..tM ) ) ;
>    ddzuv  := (u,v) -> evalf( int(int( Z(u-s,v-t) * ddphiuv
(s,t), s=tm..tM), t=tm..tM ) ) ;
>
```

```
>    x0     := x  (q,p) ;
>    y0     := y  (q,p) ;
>    z0     := z  (q,p) ;
>
>    dxu0   := dxu (q,p) ;
>    dxv0   := dxv (q,p) ;
>    dyu0   := dyu (q,p) ;
>    dyv0   := dyv (q,p) ;
>    dzu0   := dzu (q,p) ;
>    dzv0   := dzv (q,p) ;
>
>    ddxu0  := ddxu(q,p) ;
>    ddxv0  := ddxv(q,p) ;
>    ddxuv0 := ddxuv(q,p) ;
>    ddyu0  := ddyu(q,p) ;
>    ddyv0  := ddyv(q,p) ;
>    ddyuv0 := ddyuv(q,p) ;
>    ddzu0  := ddzu(q,p) ;
>    ddzv0  := ddzv(q,p) ;
>    ddzuv0 := ddzuv(q,p) ;
>
>    tg     := linalg[matrix](2,2,[evalf (dzu0 / dxu0), evalf(dzv0
/ dxv0), evalf(dzu0 / dyu0), evalf(dzv0 / dyv0)]);
>
>    crv    := linalg[matrix](3,3,[ddxu0, ddxuv0, ddxv0, ddyu0,
ddyuv0, ddyv0,ddzu0, ddzuv0, ddzv0]);
>
>
>    print( "At ", x0,y0,z0, " tg=", tg , "crv=", crv) ;
>
>   end proc :
```

B.0.8
Results

Basic test function

phie, dphieu, dphiev, ddphieu, ddphiev, ddphieuv := TFonction([1],[0],[0]) ; convole(phie, dphieu, dphiev, ddphieu, ddphiev, ddphieuv, 0, 0) ;

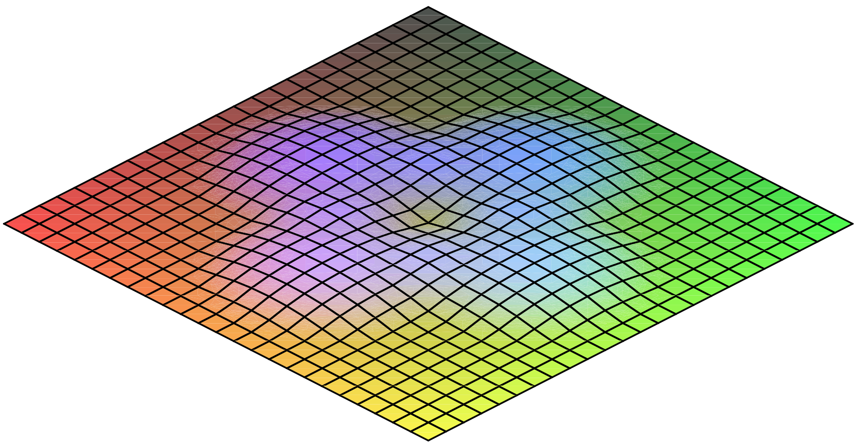


phie, dphieu, dphiev, ddphieu, ddphiev, ddphieuv := fe, dfeu, dfev, ddfeu, ddfev, ddfeuv

Error, (in evalf/int) unable to convert to pwlist

Positive and axis symmetric test function

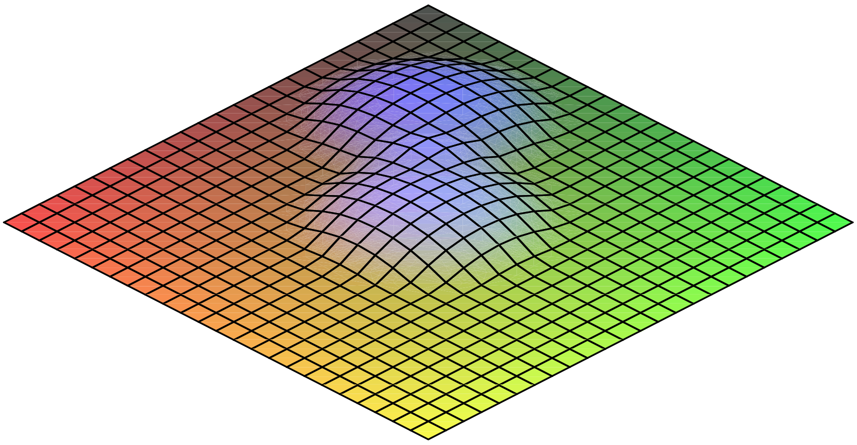
phie_sym, dphieu_sym, dphiev_sym, ddphieu_sym, ddphiev_sym, ddphieuv_sym := TFonction([1,1,1,1],[1,0,-1,0],[0,1,0,-1]) ; convole(phie_2sym, dphie_2sym, ddphie_2sym,phie_2sym, dphie_2sym, ddphie_2sym,0, 0) ;



e_sym, dphieu_sym, dphiev_sym, ddphieu_sym, ddphiev_sym, ddphieuv_sym := fe, dfeu, dfev, ddfeu, ddfev, ddfeuv

Positive and non-symmetric test function

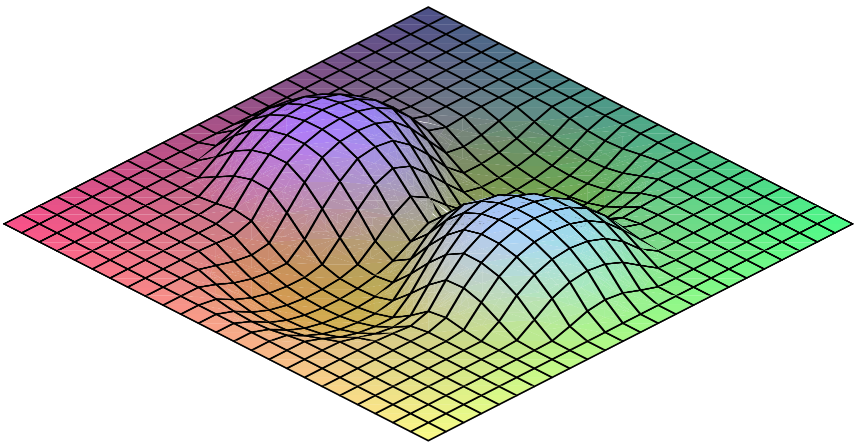
phie_dis, dphieu_dis, dphiev_dis, ddphieu_dis, ddphiev_dis, ddphieuv_dis := TFonction([1,1], [0,+1], [0,1]) : convole(phie_dis, dphieu_dis, dphiev_dis, ddphieu_dis, ddphiev_dis, ddphieuv_dis, 0, 0) ;



Error, (in evalf/int) unable to convert to pwlist

Non-positive and axis symmetric test function

phie_nsym, dphieu_nsym, dphiev_nsym, ddphieu_nsym, ddphiev_nsym, ddphieuv_nsym := TFonction([2,2,-1,-1],[1,-1,0,0],[0,0,1,-1]) ; convole(phie_nsym, dphieu_nsym, dphiev_nsym, ddphieu_nsym, ddphiev_nsym, ddphieuv_nsym,0, 0) ;

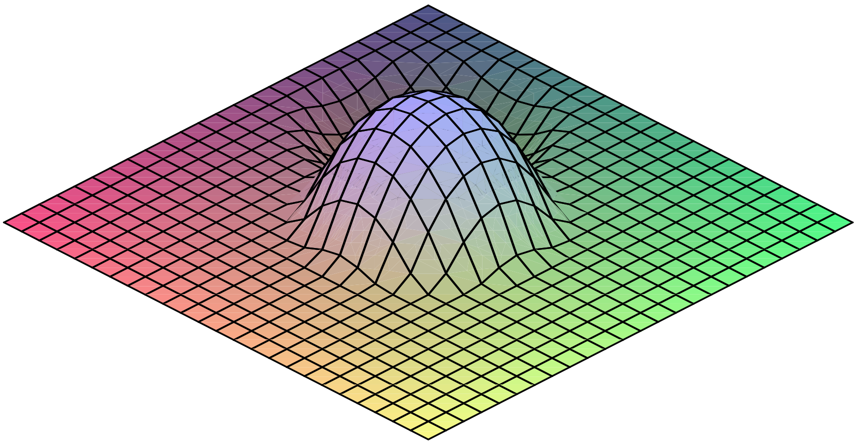


sym, dphieu_nsym, dphiev_nsym, ddphieu_nsym, ddphiev_nsym, ddphieuv_nsym := fe, dfeu, dfev, ddfeu, ddfev, ddfe

Error, (in evalf/int) unable to convert to pwlist

Non positive and non-symmetric test function

phie_ndis, dphieu_ndis, dphiev_ndis, ddphieu_ndis, ddphiev_ndis, ddphieuv_ndis := TFonction([2,-1], [0,+1],[0,1]) ; convole(phie_ndis, dphieu_ndis, dphiev_ndis, ddphieu_ndis, ddphiev_ndis, ddphieuv_ndis,0, 0) ;



$e\_ndis, dphieu\_ndis, dphiev\_ndis, ddphieu\_ndis, ddphiev\_ndis, ddphieuv\_ndis := fe, dfeu, dfev, ddfeu, ddfev, ddfeuv$

Error, (in evalf/int) unable to convert to pwlist

>