

UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE INFORMÁTICA

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Nivan Roberto Ferreira Júnior

Implementação de Rotinas para Cálculo de Conexões e Curvaturas

Recife

2009

Nivan Roberto Ferreira Júnior

Implementação de Rotinas para Cálculo de Conexões e Curvaturas

Monografia apresentada ao Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do Grau de Bacharel em Ciência da Computação.

Orientadora: Prof. Dra. Marcilia Andrade Campos

Co-Orientador: Prof. Dr. Ramón Oreste Mendoza Ahumada

Recife

2009

Agradecimentos

Primeiramente, gostaria de agradecer a Deus pela vida, pela família que tenho, pelas pessoas que encontrei e por ter me conduzido por um longo caminho até aqui.

Gostaria de agradecer a minha família, meu pai Nivan, minha mãe Fátima e meu irmão Marcelo, por ter me transformado no que sou hoje. Pelo apoio que me foi concedido durante toda minha vida e, em particular, durante esses longos 3 anos e meio de graduação.

À minha orientadora professora Marcília Andrade Campos, pela confiança, apoio e conselhos que muito me motivaram na realização deste trabalho e na continuação dos meus estudos.

Ao professor e amigo Ramón Mendoza pela recomendação para esse trabalho, por toda a ajuda que me deu para a realização dele e de muitos outros como a participação na OBM-NU, a realização das disciplinas de mestrado no DMAT-UFPE e pela orientação.

À professora Jacqueline Rojas que com seu minucioso trabalho de revisão e esclarecimentos também forneceu grande ajuda para a realização deste trabalho. Aos membros do Centro de Informática, funcionários, professores e colegas estudantes, que me acompanharam durante toda a graduação.

Finalmente, gostaria de agradecer aos membros do Departamento de Matemática da UFPE (professores, alunos e funcionários) por terem me recebido muito bem e pelo apoio. Em especial, gostaria de agradecer ao professor Francisco Brito, por ter me colocado

em contato com o DMat e pela orientação nos meus primeiros passos como estudante de matemática, e ao professor Miguel Loayza, meu orientador de iniciação científica, com quem muito aprendi.

Resumo

Conexões e Curvatura são importantes conceitos algébricos (tendo origem na geometria), que além de serem aplicados em pesquisas matemáticas, são amplamente utilizados em áreas como Física Quântica, Relatividade e Química. O uso de conexões e curvatura em aplicações requer uma quantidade grande de computações, fato este que por muitas vezes limita a complexidade das situações que podem ser investigadas por meio de computações manuais. O principal objetivo deste trabalho é a implementação de rotinas computacionais para o cálculo de conexões e curvatura. Além disso, este trabalho tem como objetivo servir como referência para o uso das rotinas implementadas por estudantes e pesquisadores, que precisem realizar computações envolvendo conexões e curvatura.

Palavras-chave: Conexões; Curvaturas; Módulos; Métrica; Computação Algébrica.

Sumário

1	Introdução	7
1.1	Motivação	7
1.2	Histórico	8
1.3	Computação Algébrica	8
2	Conceitos Matemáticos	10
2.1	Derivações e 1-Formas diferenciais	10
2.2	Conexões	12
2.2.1	A conexão de Levi-Civita	13
2.3	Operador Curvatura	15
3	Implementação	18
3.1	Planejamento da Implementação	19
3.2	Derivações	19
3.3	Conexões	22
3.3.1	A conexão de Levi-Civita	23

3.3.2	Operador Curvatura	27
4	Conclusões e Trabalhos Futuros	34
5	Manual de Instalação	35
6	Lista de Procedimentos	36

Capítulo 1

Introdução

Neste capítulo, descreve-se, de maneira geral, o trabalho realizado. Na Seção 1.1, apresenta-se a motivação do projeto. Na Seção 1.2, descreve-se o desenvolvimento histórico dos conceitos de Conexões e Curvatura. Na Seção 1.3, apresenta-se de maneira geral os conceitos da área de computação algébrica, que é utilizada como ferramenta para a implementação das rotinas para cálculo de conexões e curvatura, que são o objetivo final deste projeto.

1.1 Motivação

Este trabalho surgiu da necessidade observada pelo professor Ramón Mendoza: dificuldades encontrada por estudantes do curso de Geometria Riemanniana do programa de doutoramento em matemática, em realizar computações que apareciam no referido curso. Tais computações exigem uma quantidade alta de cálculos que facilmente induzem a erros devido a complicações com notações, conceitos e índices. Por ainda não totalmente difundido entre os estudantes de ciências básicas, sobre tudo em matemática, o uso do computador como ferramenta de auxílio a resolução de problemas, surgiu a grande motivação desse trabalho: Implementar rotinas que realizem as computações referidas e, além disso, tentar mostrar a necessidade do uso de ferramentas computacionais em estudos de ciências básicas.

1.2 Histórico

Os primórdios da geometria diferencial surgiram com Gauss (Braunschweig, 30 de Abril de 1777 - Göttingen, 23 de Fevereiro de 1855). O príncipe dos matemáticos introduziu diversos conceitos geométricos importantes (como, por exemplo, a curvatura de Gauss), quando tentava solucionar o problema de medir a extensão territorial de um reino, que lhe foi proposto pelo rei de seu país. Os conceitos introduzidos por Gauss, foram generalizados por muitos matemáticos, destacando-se entre eles Georg Friedrich Bernhard Riemann (Breselenz, Reino de Hanover, 17 de Setembro de 1826 - Selasca, Verbania, 20 de Junho de 1866). Este grande matemático foi o fundador da chamada Geometria Riemanniana, de onde surgiram os objetos matemáticos que são o foco deste trabalho: Conexões e Curvatura. O conceito de conexão surgiu dos trabalhos de Elwin Bruno Christoffel (November 10, 1829 in Montjoie, now called Monschau - March 15, 1900 in Strasbourg) com a criação da chamada conexão de Christoffel. Posteriormente, as conexões foram estudadas por Tullio Levi-Civita (Padua, 29 de março de 1873 - Roma, 29 de dezembro de 1941). Este último foi de fundamental importância para o desenvolvimento de conexões, demonstrando a existência da conexão de Levi-Civita e o importante processo de transporte paralelo (que consiste em uma generalização de translações paralelas no plano). Além disso, Levi-Civita explorou a relação entre o transporte paralelo e a curvatura para desenvolver a noção moderna de holonomia.

1.3 Computação Algébrica

A necessidade do uso de computador para realizar computações matemáticas simbólicas surge naturalmente sempre que se precisa trabalhar com computações longas e tediosas. Por exemplo, é comum um estudante ou um pesquisador gastar dias ou até mesmo semanas trabalhando em uma manipulação algébrica. Tal manipulação poderia, por exemplo, ser a solução de um sistema linear de equações ou uma integral imprópria. A área de computação algébrica tenta prover ferramentas para resolver tais problemas. Essa área consiste na criação de algoritmos computacionais capazes de realizar computações simbólicas, precisão arbitrária de números entre outras atividades. Esta área vem ganhando grande atenção nos últimos tempos devido ao constante aumento no uso

de sistemas de computação algébrica na comunidade científica [3].

O restante deste trabalho está organizado da seguinte maneira: No Capítulo 2, são apresentados os conceitos matemáticos básicos necessários para o entendimento e a fundamentação das rotinas implementadas. No Capítulo 3, descreve-se a implementação propriamente dita e seu uso é exemplificado por meio de exercícios propostos em [2] e exemplos presentes em [1].

Capítulo 2

Conceitos Matemáticos

No presente capítulo, serão apresentados definições e resultados necessários para a implementação feita. Na Seção 2.1, daremos as definições de álgebras sobre um corpo e derivações, conceitos esses que servem de base para tudo que se segue. Na Seção 2.2, será apresentado o conceito de Conexão e serão mostrados alguns resultados que servirão de base para a implementação de cálculos com conexões e também serão discutidas métricas Riemannianas e a conexão de Levi-Civita. Na Seção 2.3, discutiremos o Operador Curvatura associado a uma conexão dada.

2.1 Derivações e 1-Formas diferenciais

Definição 2.1. *Seja $\mathcal{A}$ uma k -álgebra. Seja $D \in \mathcal{L}(\mathcal{A})$ (operadores k -lineares de $\mathcal{A}$), diz-se que D é uma derivação, se D satisfaz:*

$$D(ab) = aD(b) + D(a)b, \forall a, b \in \mathcal{A}$$

O conjunto das derivações de $\mathcal{A}$ será denotado por $Der(\mathcal{A})$.

Exemplo 2.1. (Derivações de $\mathbb{R}^n$) É fácil verificar que o conjunto $C_{\mathbb{R}^n}^\infty$, das funções infinitamente diferenciáveis definidas em $\mathbb{R}^n$ e tomando valores reais é uma $\mathbb{R}$ -álgebra, com as operações de soma e multiplicação usuais (ponto a ponto). Também é de fácil verificação que as derivadas parciais $\{\partial_1, \dots, \partial_n\}$ são derivações em $C_{\mathbb{R}^n}^\infty$. De fato, uma vez

que tenhamos definida a noção de módulo, as derivadas parciais formam uma base para o $C_{\mathbb{R}^n}^\infty$ -módulo das derivações de $C_{\mathbb{R}^n}^\infty$.

Definição 2.2. Uma aplicação $f : \mathcal{M} \rightarrow \mathcal{N}$ entre $\mathcal{A}$ -módulos $\mathcal{M}$ e $\mathcal{N}$ é dita um homomorfismo de $\mathcal{A}$ -módulos, se f é um homomorfismo de grupos abelianos e for linear com respeito ao anel $\mathcal{A}$. Em particular, o conjunto dos homomorfismos de $\mathcal{A}$ -módulos $f : \mathcal{M} \rightarrow \mathcal{A}$ é denominado o dual de $\mathcal{M}$ e é denotado por $\mathcal{M}^*$. Finalmente, os elementos de $(\text{Der}(\mathcal{A}))^*$ são denominados 1-formas diferenciais.

Definição 2.3. Dado uma anel $\mathcal{A}$, um grupo abeliando $\mathcal{M}$ é dito um $\mathcal{A}$ -módulo, se nele estiver definida uma operação de multiplicação por escalar $\cdot : \mathcal{A} \times \mathcal{M} \rightarrow \mathcal{M}$ tal que para todos $r, s \in \mathcal{A}$ e $x, y \in \mathcal{M}$, verifica-se:

$$(i) \quad r \cdot (x + y) = r \cdot x + r \cdot y$$

$$(ii) \quad (r + s) \cdot x = r \cdot x + s \cdot x$$

$$(iii) \quad (rs) \cdot x = r \cdot (s \cdot x)$$

$$(iv) \quad 1_{\mathcal{A}} \cdot x = x$$

onde $1_{\mathcal{A}}$ denota a identidade do anel $\mathcal{A}$.

Exemplo 2.2. Seja $\mathcal{A}$ é uma k -álgebra com unidade. Com as operações de soma e multiplicação por um elemento de $\mathcal{A}$ definidas ponto a ponto, $\text{Der}(\mathcal{A})$ é um $\mathcal{A}$ -módulo.

Definição 2.4. Dada uma base $\Sigma = \{s_\alpha | \alpha \in \Lambda\}$ de um $\mathcal{A}$ -módulo $\mathcal{M}$, define-se a base dual de Σ como sendo o conjunto $\gamma = \{f_\alpha \in \mathcal{M}^* | \alpha \in \Lambda\}$, tal que $f_\alpha(s_\beta) = \delta_\beta^\alpha$, onde δ denota o delta de Kronecker.

Definição 2.5. Seja V um k -espaço vetorial. V é uma k -álgebra de Lie, se existir uma operação binária $[\cdot, \cdot] : V \times V \rightarrow V$, com as seguintes propriedades:

$$(i) \quad [X, Y] = -[Y, X], \quad \forall X, Y \in V$$

$$(ii) \quad [\lambda_1 X_1 + \lambda_2 X_2, Y] = \lambda_1 [X_1, Y] + \lambda_2 [X_2, Y] \text{ e } [X, \lambda_1 Y_1 + \lambda_2 Y_2] = \lambda_1 [X, Y_1] + \lambda_2 [X, Y_2], \\ \forall X, Y, X_1, X_2, Y_1, Y_2 \in V \text{ e } \forall \lambda_1, \lambda_2 \in k$$

Exemplo 2.3. Com o comutador $[D_1, D_2] = D_1 \circ D_2 - D_2 \circ D_1$, $\text{Der}(\mathcal{A})$ é uma k -álgebra de Lie.

2.2 Conexões

Nesta seção, defini-se conexões em geral, mostra-se como calcular uma conexão no caso em que os módulos são livres de posto finito. Além disso, define-se a conexão de Levi-Civita.

Definição 2.6. *Seja $\mathcal{A}$ uma $\mathbb{R}$ -álgebra e $\mathcal{M}$ um $\mathcal{A}$ -módulo. Uma conexão é uma aplicação,*

$$\begin{aligned}\nabla : \text{Der}(\mathcal{A}) \times \mathcal{M} &\longrightarrow \mathcal{M} \\ (X, s) &\longrightarrow \nabla(X, s) \equiv \nabla_X s\end{aligned}$$

que satisfaz as seguintes propriedades:

- (i) $\nabla_{X+Y}s = \nabla_X s + \nabla_Y s$
- (ii) $\nabla_{\alpha X}s = \alpha \nabla_X s$, ou seja, $\nabla(\cdot, s) : \text{Der}(\mathcal{A}) \longrightarrow \mathcal{M}$ é $\mathcal{A}$ linear;
- (iii) $\nabla_X(s_1 + s_2) = \nabla_X s_1 + \nabla_X s_2$
- (iv) (Regra de Leibntz generalizada) $\nabla_X(\alpha s) = X(\alpha)s + \alpha \nabla_X s$, ou seja, $\nabla_X(\cdot)$ é k -linear

$\forall \alpha \in \mathcal{A}$ e s, s_1 e $s_2 \in \mathcal{M}$.

A proposição a seguir apresenta uma caracterização de conexões em módulos livres de posto finito, que será de grande importância para a implementação do cálculo de conexões.

Proposição 2.1. *Seja $\mathcal{A}$ uma $\mathbb{R}$ -álgebra, $\mathcal{M}$ um $\mathcal{A}$ -módulo livre de posto N . Suponhamos que o módulo das derivações de $\mathcal{A}$, $\text{Der}(\mathcal{A})$, também seja livre, de posto n . Fixada uma base Σ de $\mathcal{M}$, existe uma bijeção $\phi_\Sigma : \text{Con}(\mathcal{M}) \rightarrow M_{N \times N}(\Omega_{\mathcal{A}}^1)$, onde $\Omega_{\mathcal{A}}^1$ denota $[\text{Der}(\mathcal{A})]^*$, o $\mathcal{A}$ -módulo das 1-formas diferenciais de $\mathcal{A}$. Explicitamente, temos que:*

$$[\nabla_D(s)]_\Sigma = \begin{pmatrix} D(s^1) \\ D(s^2) \\ \dots \\ D(s^N) \end{pmatrix} + \begin{pmatrix} \omega_1^1(D) & \omega_2^1(D) & \omega_3^1(D) & \dots & \omega_N^1(D) \\ \omega_1^2(D) & \omega_2^2(D) & \omega_3^2(D) & \dots & \omega_N^2(D) \\ \dots & \dots & \dots & \dots & \dots \\ \omega_1^N(D) & \omega_2^N(D) & \omega_3^N(D) & \dots & \omega_N^N(D) \end{pmatrix} \cdot \begin{pmatrix} s^1 \\ s^2 \\ \dots \\ s^N \end{pmatrix}$$

Da proposição, concluímos que para representar uma conexão (fixada uma base Σ do $\mathcal{A}$ -módulo $\mathcal{M}$) é necessário armazenar uma matriz de 1-formas diferenciais e, por sua vez, fixada uma base β do módulo $Der(\mathcal{A})$, estas 1-formas diferenciais são representadas por matrizes de linhas, cuja dimensão é a dimensão do módulo das derivações de $\mathcal{A}$. Ainda da proposição, obtemos uma maneira de calcular o valor da conexão em uma derivação D , cujo vetor de coordenadas na base β é representado por $[D]_\beta$, e em um elemento s de $\mathcal{M}$, cujo vetor de coordenadas na base Σ é representado pela equação acima.

2.2.1 A conexão de Levi-Civita

Nesta seção será definido o importante conceito de métrica e será mostrado como, a partir de uma métrica dada, associar a ela uma conexão especial chamada a conexão de Levi-Civita associada à métrica. Serão dadas definições breves. Para exemplos e mais informações sobre métricas e/ou provas dos teoremas apresentados aqui, o leitor deve consultar a Seção 2.1 [1].

Definição 2.7. *Seja $\mathcal{A}$ uma $\mathbb{R}$ -álgebra. Uma métrica g é uma aplicação bilinear*

$$g : Der(\mathcal{A}) \times Der(\mathcal{A}) \longrightarrow \mathcal{A}$$

que satisfaz a seguinte propriedade: A função

$$\begin{aligned} \psi : Der(\mathcal{A}) &\longrightarrow (Der(\mathcal{A}))^* \\ X &\longrightarrow \psi(X) = g(*, X) \end{aligned}$$

é um isomorfismo.

Definição 2.8. *Dada uma conexão ∇ ,*

$$\begin{aligned} \tau^\nabla : Der(\mathcal{A}) \times Der(\mathcal{A}) &\longrightarrow Der(\mathcal{A}) \\ (X, Y) &\longrightarrow \tau^\nabla(X, Y) = \nabla_X Y - \nabla_Y X - [X, Y] \end{aligned}$$

é chamada a torção da conexão ∇ .

Diz-se que a conexão ∇ é simétrica, se $\forall X, Y \in Der(\mathcal{A})$ tivermos que a torção $\tau^\nabla \equiv 0$, ou seja,

$$\nabla_X Y - \nabla_Y X = [X, Y] \tag{2.1}$$

Definição 2.9. Dizemos que a conexão ∇ é compatível com a métrica g se $\forall X, Y, Z \in \text{Der}(\mathcal{A})$ tivermos

$$X(g(Y, Z)) = g(\nabla_X Y, Z) + g(Y, \nabla_X Z) \quad (2.2)$$

Teorema 2.1. (Levi-Civita) Dada uma métrica g , existe uma única conexão denotada por $\nabla^g \in \text{Con}(\text{Der}(\mathcal{A}))$ que possui as seguintes propriedades:

(i) ∇^g é simétrica

(ii) $X(g(Y, Z)) = g(\nabla_X Y, Z) + g(Y, \nabla_X Z), \forall X, Y, Z \in \text{Der}(\mathcal{A})$.

Demonstração. Daremos aqui um esboço da prova deste teorema, para maiores detalhes ver, por exemplo, [1] ou [2].

Caso tal conexão existisse, ela deveria satisfazer

$$X(g(Y, Z)) = g(\nabla_X^g Y, Z) + g(Y, \nabla_X^g Z) \quad (2.3)$$

$$Y(g(Z, X)) = g(\nabla_Y^g Z, X) + g(Z, \nabla_Y^g X) \quad (2.4)$$

$$Z(g(X, Y)) = g(\nabla_Z^g X, Y) + g(X, \nabla_Z^g Y) \quad (2.5)$$

somando (1.3) e (1.4) e subtraindo (1.5) obtem-se,

$$\begin{aligned} X(g(Y, Z)) + Y(g(Z, X)) - Z(g(X, Y)) &= g(\nabla_X^g Y, Z) + g(Y, \nabla_X^g Z) + g(\nabla_Y^g Z, X) \\ &\quad + g(Z, \nabla_Y^g X) - g(\nabla_Z^g X, Y) - g(X, \nabla_Z^g Y) \end{aligned}$$

somando e subtraindo o termo $g(\nabla_Y^g X, Z)$ temos,

$$\begin{aligned} X(g(Y, Z)) + Y(g(Z, X)) - Z(g(X, Y)) &= g(\nabla_X^g Y, Z) + g(Y, \nabla_X^g Z) + g(\nabla_Y^g Z, X) \\ &\quad + g(Z, \nabla_Y^g X) - g(\nabla_Z^g X, Y) - g(X, \nabla_Z^g Y) \\ &\quad + (\nabla_Y^g X, Z) - g(\nabla_Y^g X, Z) \\ &= g(Y, \nabla_X^g Z) - g(\nabla_Z^g X, Y) + g(\nabla_Y^g Z, X) \\ &\quad - g(X, \nabla_Z^g Y) + g(\nabla_X^g Y, Z) - g(\nabla_Y^g X, Z) \\ &\quad + 2g(Z, \nabla_Y^g X) \end{aligned}$$

Assim,

$$\begin{aligned} g(Z, \nabla_Y^g X) &= \frac{1}{2} \{X(g(Y, Z)) + Y(g(Z, X)) - Z(g(X, Y)) \\ &\quad - g(Y, [X, Z]) - g(X, [Y, Z]) - g(Z, [X, Y])\} \end{aligned}$$

Fixados X e Y , defina

$$\begin{aligned} \psi(Z) &= \frac{1}{2} \{X(g(Y, Z)) + Y(g(Z, X)) - Z(g(X, Y)) \\ &\quad - g(Y, [X, Z]) - g(X, [Y, Z]) - g(Z, [X, Y])\} \end{aligned}$$

Verifica-se que ψ é um funcional linear e que a conexão desejada é a representação do funcional, no sentido da definição (1.6), ou seja,

$$\psi(Z) = g(Z, \nabla_Y^g X), \forall Z \in \text{Der}(\mathcal{A})$$

Observação 2.1. Se $\mathcal{U}$ for uma variedade diferenciável, dois exemplos importantes são as métricas riemannianas e as métricas lorentzianas no $C_{\mathcal{U}}^{\infty}$ -módulo das seções do fibrado tangente de $\mathcal{U}$.

2.3 Operador Curvatura

Definição 2.10. *Seja $\mathcal{A}$ uma $\mathbb{R}$ -álgebra e $\mathcal{M}$ um $\mathcal{A}$ -módulo. O operador curvatura associado a uma conexão ∇ é uma aplicação,*

$$\begin{aligned} K^{\nabla} : \text{Der}(\mathcal{A}) \times \text{Der}(\mathcal{A}) \times \mathcal{M} &\longrightarrow \mathcal{M} \\ (X, Y, s) &\longrightarrow K^{\nabla}(X, Y, s) \end{aligned}$$

definida por

$$K^{\nabla}(X, Y, s) = [\nabla_X, \nabla_Y] s - \nabla_{[X, Y]} s \quad (2.6)$$

Onde $\nabla_X, \nabla_Y \in \text{End}_k(\mathcal{M})$.

Definição 2.11. *Uma função $\omega : \text{Der}(\mathcal{A}) \times \text{Der}(\mathcal{A}) \rightarrow \mathcal{A}$ é dita uma 2-forma diferencial, se:*

(i) ω é linear em cada variável

(ii) ω é alternante, i.e., $\omega(X, Y) = -\omega(Y, X)$, $\forall X, Y \in \text{Der}(\mathcal{A})$

O conceito de 2-forma diferencial mostrar-se-á importante no estudo do operador curvatura.

No caso em que $\mathcal{M}$ é um $\mathcal{A}$ -módulo livre de posto finito N e de base $\{s_\alpha\}_{\alpha=1}^N$, pelo que vimos no capítulo anterior podemos definir o operador curvatura na base do módulo. Se definirmos

$$K^\nabla(X, Y, s_\alpha) = \sum_{\beta=1}^n K_\alpha^\beta(X, Y) s_\beta = ([\nabla_X, \nabla_Y] - \nabla_{[X, Y]}) s_\alpha, \quad (2.7)$$

para cada K_α^β temos:

Proposição 2.2. *As aplicações $K_\alpha^\beta : \text{Der}(\mathcal{A}) \times \text{Der}(\mathcal{A}) \longrightarrow \mathcal{A}$ satisfazem as seguintes propriedades:*

(i) $K_\alpha^\beta(X, Y) = -K_\alpha^\beta(Y, X)$;

(ii) $K_\alpha^\beta(\lambda X, Y) = \lambda K_\alpha^\beta(X, Y)$ com $\lambda \in \mathcal{A}$.

Demonstração. Ver proposição 3.1 de [1].

■

Assim como foi feito para conexões, a próxima proposição visa a associar ao operador curvatura uma matriz. Utiliza-se na referida proposição derivada e produto exteriores. Para referência sobre tais conceitos ver, por exemplo, [6].

Proposição 2.3. *Seja $\mathcal{M}$ um $\mathcal{A}$ -módulo livre de posto finito N e $\Sigma = \{s_1, \dots, s_N\}$ uma base de $\mathcal{M}$. Dada uma conexão ∇ , cuja matriz na base α é $[(\omega^\nabla)_\alpha]^\beta = \omega$, associamos a curvatura K^∇ a matriz $[(K^\nabla)_\alpha]^\beta = K$, esta é dada por:*

$$K = d\omega + \omega \wedge \omega \quad (2.8)$$

isto é, para cada entrada da forma curvatura temos,

$$K_\alpha^\beta = d\omega_\alpha^\beta + \sum_{k=1}^N \omega_\gamma^\beta \wedge \omega_\alpha^\gamma \quad (2.9)$$

Demonstração. Ver Proposição 3.3 de [1].



Capítulo 3

Implementação

Neste capítulo, será apresentada a implementação feita para o cálculo de conexões e curvatura. Para tal foi utilizado o Maple 12. Como o foco é em aspectos computacionais, serão tratados apenas $\mathcal{A}$ -módulos livres de posto finito, onde $\mathcal{A}$ denota uma $\mathbb{R}$ -álgebra, será suposto também que o módulo das derivações da $\mathbb{R}$ -álgebra $\mathcal{A}$ também é livre de posto finito. Assim, em tudo que se segue tem-se fixadas bases $\beta = \{\partial_1, \dots, \partial_n\}$ e $\Sigma = \{s_1, \dots, s_N\}$, das derivações e do $\mathcal{A}$ -módulo $\mathcal{M}$. Será feita a suposição de que os elementos de β são as derivadas parciais. Este é um caso importante, conforme [5].

Este trabalho não tem o objetivo de ser uma introdução ao uso do Maple, para tal fim existem vários livros e tutoriais disponíveis, dos quais citamos [4].

Na Seção 3.1, faz-se considerações sobre softwares de computação algébrica e outras possíveis opções de ferramentas para a implementação feita; além disso, apresentam-se as razões para escolha do Maple como ferramenta para a implementação das rotinas tema deste trabalho. Na Seção 3.2, apresenta-se a implementação dos conceitos da Seção 2.1. Similarmente, a Seção 3.3 apresenta-se a implementação da Seção 2.2 referentes a conexões e, finalmente, na Seção 3.3.2 descreve-se a implementação da Seção 2.3

3.1 Planejamento da Implementação

Na etapa de planejamento da implementação, questões relativas ao escopo da implementação e qual tecnologia utilizar foram consideradas. As decisões foram tomadas principalmente considerando questões relativas a tempo de realização do projeto e público alvo. Quanto ao escopo da implementação, decidiu-se pela implementação do estudo feito em [1]. Tal decisão foi devida ao fato de o material em [1] ser bastante acessível. Além disso, [1] contém resultados com fins de possibilitar implementações de conexões e curvaturas. Quanto a tecnologia utilizada, foram consideradas diversas alternativas considerando softwares livres e ferramentas corporativas e a tecnologia que melhor satisfizes as restrições foi o Maple, a ferramenta escolhida para a implementação.

3.2 Derivações

Nesta Seção será ilustrado como representar uma derivação e como calcular o valor de uma derivação D , cujas coordenadas na base α são $[D]_\alpha = [D^1, D^2, \dots, D^n]$, em uma função f . Nas implementações das funções que se seguem a variável $\dim Der$ denotará a dimensão do módulo das derivações, ou seja, $\dim Der = n$. O objetivo aqui é, a partir de, D fazer a seguinte computação:

$$D(f) = \sum_{1 \leq i \leq n} D^i \partial_i(f)$$

A implementação é feita pela função `calcularDerivacaoFuncao`, cuja implementação é dada a seguir:

```

> calcularDerivacaoFuncao := proc(der,func)::function;
> local i, f,deri;
> f:=0;

> deri := Matrix(der);
> for i from 1 to dimDer do

> f:= f + deri[1,i]*diff(func,variaveis[i]);
> od;

> return f;
> end:

```

Como exemplo, calculamos a derivação $D = [\sin(x_1), \cos(x_1)]$ na função $f(x_1, x_2) = x_1 * x_2$. A seguir está a saída no maple:

```

> calcularDerivacaoFuncao([sin(x1),cos(x1)], x1*x2);
sin(x1) x2 + cos(x1) x1

```

Usa-se a função *calcularDerivacaoFuncao* para implementar a função *calcularDerivacao* que recebe uma derivação D , com as mesmas coordenadas citadas acima e um elemento f do $\mathcal{A}$ -módulo $\mathcal{M}$, cujas coordenadas, na base, β são $[f]_\beta = [f^1, \dots, f^N]$ e realiza o seguinte cálculo:

$$calcularDerivacao(D, f) = \begin{pmatrix} D(f^1) \\ D(f^2) \\ \dots \\ D(f^N) \end{pmatrix} = \begin{pmatrix} \sum_{1 \leq i \leq n} D^i \partial_i(f^1) \\ \sum_{1 \leq i \leq n} D^i \partial_i(f^2) \\ \dots \\ \sum_{1 \leq i \leq n} D^i \partial_i(f^N) \end{pmatrix}$$

A implementação do procedimento é a seguinte:

```

> calcularDerivacao := proc(der, sec)
> local i, newSec,sect,deri;

> sect := Matrix(sec);

> deri := Matrix(der);

> newSec := Matrix(1,dimSec);

> for i from 1 to dimSec do

```

```

> newSec[1,i] := calcularDerivacaoFuncao(deri,sect[1,i]);
> od;

> return newSec;
> end:

```

Como anteriormente, calcula-se a derivação $D = [\sin(x_1), \cos(x_2)]$ no elemento $f = [x_1 * x_2, x_1^2]$ de $\mathcal{M}$. A seguir está a saída no maple:

```

> calcularDerivacao([sin(x1),cos(x2)],[x1*x2, x1*x1]);

```

$$\begin{bmatrix} \sin(x_1) x_2 + \cos(x_2) x_1 & 2 \sin(x_1) x_1 \end{bmatrix}$$

Assim como as derivações, as 1-formas diferenciais (elementos de $(Der(\mathcal{A}))^*$) são representadas por vetores de elementos de $\mathcal{A}$. A função *calcular1FormaDerivacao*, retorna o valor de uma 1-forma diferencial em uma derivação dada. Simbolicamente, se ω é uma 1-forma diferencial tal que $[\omega]_\gamma = [\omega_1, \dots, \omega_n]$ e D é uma derivação, cujas coordenadas na base α são $[D]_\alpha = [D^1, \dots, D^n]$, temos que

$$calcular1FormaDerivacao(\omega, D) = \sum_{1 \leq i \leq n} \omega_i D^i,$$

supondo que γ é a base dual de α . A implementação do procedimento é dada a seguir:

```

> calcular1FormaDerivacao := proc(forma, der)
> local i, result;
> result := 0;
> for i from 1 to dimDer do
> result := result + forma[i]*der[i];
> od;
> return result;
> end:

```

Como exemplo de uso de *calcular1FormaDerivacao* para calcular o valor da 1-forma δ tal que $[\delta]_\gamma = [x_1 * x_2, x_1 * \sec(x_2)]$ em uma derivação D tal que $[D]_\beta = [\cos(x_1 * x_2), x_2^3]$.

```
> calcular1FormaDerivacao([x1*x2,x^1*sex(x2)], [cos(x1*x2),x2^3]);
x1 x2 cos(x1 x2) + x sex(x2) x2^3
```

3.3 Conexões

A fim de representar uma conexão ∇ , segundo a proposição 2.1, é necessário apenas guardar a matriz da conexão em uma determinada base do módulo $\mathcal{M}$. Ainda da proposição 2.1, obtemos uma maneira de calcular o valor da conexão em uma derivação D , cujo vetor de coordenadas na base β é representado por $[D]_\beta$, e em um elemento s de $\mathcal{M}$, cujo vetor de coordenadas na base Σ é representado por $[s]_\Sigma$:

$$[\nabla_D(s)]_\Sigma = \begin{pmatrix} D(s^1) \\ D(s^2) \\ \dots \\ D(s^N) \end{pmatrix} + \begin{pmatrix} \omega_1^1(D) & \omega_2^1(D) & \omega_3^1(D) & \dots & \omega_N^1(D) \\ \omega_1^2(D) & \omega_2^2(D) & \omega_3^2(D) & \dots & \omega_N^2(D) \\ \dots & \dots & \dots & \dots & \dots \\ \omega_1^N(D) & \omega_2^N(D) & \omega_3^N(D) & \dots & \omega_N^N(D) \end{pmatrix} \cdot \begin{pmatrix} s^1 \\ s^2 \\ \dots \\ s^N \end{pmatrix}$$

O procedimento que realiza o cálculo de uma conexão ∇ dada em uma derivação D e um elemento s do módulo $\mathcal{M}$ é *calcularConexao*. A implementação é dada abaixo

```
> calcularConexao := proc(matrizForma, der, sec)
> local con;
> con :=
> MatrixAdd(calcularDerivacao(der,sec),MatrixMatrixMultiply(Matrix(calcu
> larMatriz1FormaDerivacao(matrizForma,der)),Matrix(transpose(zeta))));
> return con;
> end;
```

A seguir, demonstra-seo uso do procedimento `calcularConexao`

```

> zeta:= Matrix([x1*x2, x1*x1]):
> t:=calcularConexao(([[ [x1 + x2, x1],[x1*3, x1+ x2]], [[x1,
> sin(x2)], [x1, x2]] ]),[x2*2 + x1,x2],zeta):
> map(simplify,t);

```

$$\begin{bmatrix} 2x^2 + 2x_1x_2 + 5x_1^2x_2^2 + 8x_1^3x_2 + 2x_1x_2^3 + 3x_1^4 \\ x_1(4x_2 + 2x_1 + 3x_1x_2^2 + 3x_1^2x_2 + x_2^2\sin(x_2) + x_1^3) \end{bmatrix}$$

3.3.1 A conexão de Levi-Civita

O principal objetivo desta Seção é implementar a função `leviCivitaMatriz` que dada uma métrica retorna a matriz da conexão de Levi-Civita associada a esta métrica, segundo o teorema 2.1. Para tanto, a estratégia será a mesma utilizada na prova do teorema 2.1: Para cada entrada $X, Y \in \text{Der}(\mathcal{A})$, obtêm-se um funcional $\psi(X, Y) \in (\text{Der}(\mathcal{A}))^*$ (definido na demonstração do teorema citado) e então encontra-se o representante deste funcional, o que resultará em $\nabla_X(Y)$. A partir de um procedimento que realiza esse cálculo, constrói-se a matriz da conexão.

O primeiro passo é calcular as coordenadas do funcional $\psi(X, Y)$ com relação a base dual de β . Tal cálculo é feito através do seguinte procedimento:

```

> calcularFuncional := proc(metrica, derX, derY)
> local result,i;
> result := Matrix(1,dimDer);
> for i from 1 to dimDer do

```



```

> result[1,i] :=
> (1/2)*(calcularDerivacaoFuncao2(derX,calcularMetrica(metrica,derY,Matr
> ix(Vector(dimDer,shape=unit[i]))))[1,1])+
> calcularDerivacaoFuncao2(derY,calcularMetrica(metrica,derX,Matrix(Vect
> or(dimDer,shape=unit[i]))))[1,1])-
> diff(calcularMetrica(metrica,derX,derY)[1,1],variaveis[i])-
> calcularMetrica(metrica,Matrix(Vector(dimDer,shape=unit[i])),comutador
> (derX,derY))[1,1]-
> calcularMetrica(metrica,derX,comutador(derY,Matrix(Vector(dimDer,shape
> =unit[i]))))[1,1]-
> calcularMetrica(metrica,derY,comutador(derX,Matrix(Vector(dimDer,shape
> =unit[i]))))[1,1]);
> od;
> return result;

```

O procedimento anterior realiza o cálculo das coordenadas de $\psi_{(X,Y)}$, calculando seu vetor nos elementos da base β . O uso dessa função será exemplificado mais adiante. Agora que o operador ψ já foi calculado é possível calcular o seu representante através de uma métrica g dada. Para isso, foi criada a função `acharRepresentanteFuncional`, descrita abaixo. Essa função toma como entrada as coordenadas de um funcional e uma métrica. O retorno é a multiplicação entre o vetor linha de coordenadas do funcional e a matriz inversa da matriz da métrica. A implementação é dada a seguir:

```

> acharRepresentanteFuncional := proc(funcional,metrica)
> local result;
> result := MatrixMatrixMultiply(Matrix(funcional),
> MatrixInverse(metrica));
> return result;
> end:

```

A implementação do procedimento anterior é baseada no fato de que, o representante de $\psi_{(X,Y)}$ é um elemento de $\nabla_X(Y) \in \text{Der}(\mathcal{A})$ tal que $\psi_{(X,Y)}(Z) = g(\nabla_X(Y), Z), \forall Z \in \text{Der}(\mathcal{A})$. Em notação matricial,

$$((\psi_{(X,Y)})_1 \quad \dots \quad (\psi_{(X,Y)})_n) \begin{pmatrix} Z_1 \\ \dots \\ Z_n \end{pmatrix} =$$

$$[(\nabla_X(Y))_1 \quad \dots \quad (\nabla_X(Y))_n] \begin{pmatrix} g_1^1 & g_2^1 & g_3^1(D) & \dots & g_n^1(D) \\ g_1^2(D) & g_2^2(D) & g_3^2(D) & \dots & g_n^2(D) \\ \dots & \dots & \dots & \dots & \dots \\ g_1^n(D) & g_2^n(D) & g_3^n(D) & \dots & g_n^n(D) \end{pmatrix} \cdot \begin{pmatrix} Z_1 \\ \dots \\ Z_n \end{pmatrix},$$

para todo $Z \in \text{Der}(\mathcal{A})$. Assumindo que $\mathcal{A}$ possui uma unidade, pode-se "cancelar" Z em ambos os lados da equação acima e obtem-se:

$$((\psi_{(X,Y)})_1 \quad \dots \quad (\psi_{(X,Y)})_n) = [(\nabla_X(Y))_1 \quad \dots \quad (\nabla_X(Y))_n] \begin{pmatrix} g_1^1 & g_2^1(D) & g_3^1(D) & \dots & g_n^1(D) \\ g_1^2(D) & g_2^2(D) & g_3^2(D) & \dots & g_n^2(D) \\ \dots & \dots & \dots & \dots & \dots \\ g_1^n(D) & g_2^n(D) & g_3^n(D) & \dots & g_n^n(D) \end{pmatrix}$$

Invertendo a matriz da métrica e multiplicando a esquerda, obtemos o resultado.

Agora, é possível calcular a conexão de Levi-Civita. A função abaixo faz esse cálculo em tomando como argumentos a métrica e derivações X e Y e retorna o valor de $\nabla_X^g(Y)$.

```
> leviCivita := proc(metrica, derX, derY)
> local i, funcional;
> for i from 1 to dimDer do
> funcional :=
> MatrixMatrixMultiply(calcularFuncional(metrica, derX, derY), Matrix(inver
> se(metrica)));
> od;
```

```
> return funcional;
> end:
```

Exemplo 3.1. Este exemplo demonstra o uso das funções descritas acima para resolver o item (a) do exercício 8 do segundo capítulo de [2]. O exercício consiste em encontrar a matriz da conexão de Levi-Civita da métrica de Lobatchevski em $\mathbb{R}^{2+} = \{(x, y) \in \mathbb{R}; y > 0\}$. As coordenadas das 1-formas diferenciais que são entradas da matriz da conexão são chamadas Símbolos de Christoffel. Tal métrica é dada pela matriz 2 por 2 g tal que $g_{11} = g_{22} = \frac{1}{y^2}$ e $g_{12} = g_{21} = 0$. Faz-se uso da função `leviCivita` dada acima, para calcular $\nabla_{\partial_1}^g(\partial_1), \nabla_{\partial_1}^g(\partial_2), \nabla_{\partial_2}^g(\partial_1), \nabla_{\partial_2}^g(\partial_2)$ estes representam as entradas da matriz da conexão. A função a seguir faz o cálculo descrito

```
> leviCivitaMatriz := proc(metric)
> local i,j,matriz;

> matriz := Array(1..dimDer,1..dimDer);
> for i from 1 to dimDer do

> for j from 1 to dimDer do
> matriz[i,j] := leviCivita(metric,
> Matrix(Vector(dimDer,shape=unit[i])),Matrix(Vector(dimDer,shape=unit[j]
> ))));
> od;
> od;
> return matriz;
> end:
```

O resultado da chamada

```
> leviCivitaMatriz([[1/(x2)^2,0],[0,1/(x2)^2]]);
```

é:

$$\begin{bmatrix} \begin{bmatrix} 0 & -\frac{1}{x^2} \\ \frac{1}{x^2} & 0 \end{bmatrix} & \begin{bmatrix} -\frac{1}{x^2} & 0 \\ 0 & -\frac{1}{x^2} \end{bmatrix} \end{bmatrix}$$

O que nos dá a solução do exercício.

Exemplo 3.2. Vamos utilizar o mesmo procedimento para calcular a matriz da conexão de Levi-Civita da métrica apresentada no exemplo 3.3 de [1]. Tal métrica é definida na esfera bidimensional S^2 da seguinte maneira: Defini-se

$$\psi : (0, 2\pi) \times (0, \pi) \rightarrow S^2$$

$$\psi(\theta, \phi) = (\sin(\phi) \cos(\theta), \sin(\phi) \sin(\theta), \cos(\phi))$$

Obtem-se (ver [1]), a partir de ϕ , as derivações $\partial_1 = \frac{\partial}{\partial \theta}$ e $\partial_2 = \frac{\partial}{\partial \phi}$ e nelas definimos uma métrica g , dada por:

$$g(\partial_1, \partial_1) = \sin^2(\phi), g(\partial_2, \partial_2) = 1, g(\partial_1, \partial_2) = g(\partial_2, \partial_1) = 0$$

A fim de calcular a conexão associada a métrica g , basta usarmos, novamente, a função `leviCivitaMatriz`:

```
> leviCivitaMatriz([[(sin(x2))^2, 0], [0, 1]]);
```

$$\begin{bmatrix} \begin{bmatrix} 0 & \frac{\cos(x2)}{\sin(x2)} \end{bmatrix} & \begin{bmatrix} \frac{\cos(x2)}{\sin(x2)} & 0 \end{bmatrix} \\ \begin{bmatrix} -\cos(x2) \sin(x2) & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 \end{bmatrix} \end{bmatrix}$$

3.3.2 Operador Curvatura

O principal objetivo desta Seção é calcular o Operador Curvatura associado a uma conexão, de acordo com a definição 2.10. Explicitamente, dada uma conexão, deseja-se calcular o valor do operador curvatura K^∇ em um ponto

$(X, Y; s) \in \text{Der}(\mathcal{A}) \times \text{Der}(\mathcal{A}) \times \mathcal{M}$ e também a matriz do operador curvatura como na proposição 2.3. Para a primeira das tarefas, temos que calcular $K^\nabla(X, Y; s) = [\nabla_X, \nabla_Y](s) - \nabla_{[X, Y]}(s)$. Na Seção anterior, o cálculo de conexões foi implementado, então basta implementar um procedimento que calcule o comutador de duas derivações. Esta implementação está descrita a seguir:

```

> comutador := proc(Z,W)
>   local i,j,result,X,Y;
>   X:=Matrix(Z);
>   Y:=Matrix(W);
>   result := Matrix(1,dimDer);
>   for j from 1 to dimDer do
>     for i from 1 to dimDer do
>       result[1,j] := result[1,j] + X[1,i]*diff(Y[1,j], variaveis[i]) -
>       Y[1,i]*diff(X[1,j], variaveis[i]);
>     od;
>   od;
>   return result;
> end:

```

Com este procedimento, a implementação do cálculo de $K^\nabla(X, Y, s)$ é:

```

> calcularOperadorCurvatura := proc(conexao, der1a, der2a, sect)
>   local retorno,der1,der2,sec;
>   der1:=Matrix(der1a);
>   der2:=Matrix(der2a);
>   sec:=Matrix(sect);
>   return calcularConexao(conexao,der1,transpose(calcularConexao(conexao,
>   der2, sec))) -
>   calcularConexao(conexao,der2,transpose(calcularConexao(conexao, der1,
>   sec))) - calcularConexao(conexao, comutador(der1,der2),sec);
> end:

```

A fim de exemplificar o uso da função *calcularOperadorCurvatura*, da conexão de Levi-Civita associada a métrica de Lobachevski (ver exemplo 3.1) no ponto $(\partial_1, \partial_2; \partial_1)$, onde ∂_1 denota o primeiro vetor da base α .

```
> calcularOperadorCurvatura([ [0, -1/x2], [-1/x2, 0]], [[1/x2, 0], [0, -1/x2]],
[1, 0], [0, 1], [1, 0]);
```

$$\begin{bmatrix} 0 \\ \frac{1}{x^2} \end{bmatrix}$$

Para calcular a matriz do operador curvatura K^∇ associado a uma conexão ∇ , segundo a proposição 2.3, faz-se necessário o cálculo de derivadas exteriores e de produto exterior de 1-formas. O resultado dessas operações é uma 2-forma diferencial. Dada uma base $\beta = \{\partial_1, \dots, \partial_n\}$ de $Der(\mathcal{A})$, obtem-se a sua base dual $\gamma = \{\delta_\beta^1, \dots, \delta_\beta^n\}$ e os vetores da forma $\delta^i \wedge \delta^j$ com $i < j$ formam uma base de $\Omega_2(\mathcal{A})$ o conjunto das duas formas diferenciais. Desse modo, para representar um elemento de $\Omega_2(\mathcal{A})$ é necessário armazenar uma matriz linha de tamanho $\frac{n(n-1)}{2}$. Será convencionado que a ordem das coordenadas será a ordem lexicográfica. Por exemplo, no caso $n = 3$, se $w \in \Omega_2(\mathcal{A})$, tem-se $w = w_{12}\delta^1 \wedge \delta^2 + w_{13}\delta^1 \wedge \delta^3 + w_{23}\delta^2 \wedge \delta^3$, com $w_{ij} \in \mathcal{A}$. Matricialmente, $[w] = [w_{12}, w_{13}, w_{23}]$. A fim de calcular o produto exterior de 2-formas diferenciais, implementou-se o seguinte procedimento:

```

> calcularProdutoExterior1Formas := proc(forma1,forma2)
> local resultado,i,j;
> resultado := Matrix(1,dimDer*(dimDer-1)/2);
> for i from 1 to dimDer*(dimDer-1)/2 do
> resultado[1,i] := 0;
> od;
> for i from 2 to dimDer do
> for j from 1 to i-1 do
> resultado[1,(i-2) + j] := resultado[1,(i-2) + j] +
> (-1)*forma1[i]*forma2[j];
> od;
> od;
> for i from 1 to dimDer-1 do
> for j from i+1 to dimDer do
> resultado[1,i + j - 2] := resultado[1,(i-2) + j] +
> forma1[i]*forma2[j];
> od;
> od;
> return resultado;
> end:

```

Para calcular a derivada exterior, implementou-se a seguinte função:

```

> calcularDerivadaExterior1Forma := proc(forma)
> local resultado,i,j;
> resultado := Matrix(1,dimDer*(dimDer-1)/2);
> for i from 1 to dimDer*(dimDer-1)/2 do
> resultado[1,i] := 0;
> od;
> for i from 2 to dimDer do
> for j from 1 to i-1 do
> resultado[1,i+j-2] := resultado[1,i+j-2] +
> (-1)*diff(forma[j],variaveis[i]);
> od;
> od;
> for i from 1 to dimDer-1 do
> for j from i+1 to dimDer do
> resultado[1,i+j-2] := resultado[1,i+j-2] +
> diff(forma[j],variaveis[i]);
> od;
> od;
> return resultado;
> end:

```

Finalmente, para calcular a matriz do operador curvatura:


```

> calcularOperadorCurvaturaMatriz := proc(conexao)
> local resultado,i,j,k,temp;
> resultado := Matrix(dimSec,dimSec);
> for i from 1 to dimSec do
> for j from 1 to dimSec do
> temp :=0;
> for k from 1 to dimSec do
> temp:= temp +
> calcularProdutoExterior1Formas(conexao[i,k],conexao[k,j]);
> od;
> resultado[i,j] := calcularDerivadaExterior1Forma(conexao[i,j])
> + temp;
> od;
> od;
> return resultado;
> end:

```

Exemplo 3.3. Como aplicação da função calcularOperadorCurvaturaMatriz, calcular-se-á a matriz do operador curvatura as conexões de Levi-Civita associadas as métricas dos exemplos 3.1 e 3.2. Para o primeiro caso:

```

> calcularOperadorCurvaturaMatriz([ [[0, -1/x2],[-1/x2, 0]], [[1/x2,
> 0],[0, -1/x2]] ]);

```

$$\begin{bmatrix} \begin{bmatrix} 0 \\ \frac{1}{x^2} \end{bmatrix} & \begin{bmatrix} -\frac{1}{x^2} \\ 0 \end{bmatrix} \\ \begin{bmatrix} \frac{1}{x^2} \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ -\frac{1}{x^2} \end{bmatrix} \end{bmatrix}$$

Para o segundo caso:

```

> calcularOperadorCurvaturaMatriz([ [0,
> cos(x2)/sin(x2)], [cos(x2)/sin(x2), 0]], [[-cos(x2)*sin(x2), 0], [0, 0]]
> ]);

```

$$\begin{bmatrix} \begin{bmatrix} 0 \\ -\sin(x^2)^2 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{bmatrix}$$

Capítulo 4

Conclusões e Trabalhos Futuros

O principal objetivo deste trabalho é prover ferramentas para estudantes e pesquisadores que utilizem os conceitos de conexões e curvaturas em suas atividades. Vemos o projeto aqui realizado como o começo de uma longa caminhada para esse fim. No presente trabalho, foram implementadas as rotinas básicas para o uso de conexões e curvatura, que apesar de sua simplicidade, fornece a possibilidade de resolvermos um grande número de problemas propostos complexos, como, por exemplo, os problemas presentes em [2]. A restrição do cálculo com o módulo das derivações sendo gerado pelas derivadas parciais foi escolhido pela aplicabilidade e pelo apelo intuitivo, mas esta restrição não constitui de maneira nenhuma um obstáculo na generalização da implementação para casos mais gerais, as idéias gerias continuam as mesmas e, devido a modularidade do código implementado, basta substituir poucas rotinas.

Como mencionado anteriormente, a grande quantidade de aplicações desses conceitos evidencia a necessidade de continuar esta implementação. Destacamos entre as principais aplicações a serem contempladas o transporte paralelo e o cálculo dos grupos de holonomia. Ambos são bastante importantes em física quântica [7]. A implementação dessas aplicações certamente tornariam o projeto aqui desenvolvido mais aplicável em contextos de pesquisa. Ressaltamos que a implementação aqui realizada constitui uma base para a continuação em muitas direções e torna a implementação das aplicações bastante viável.

Capítulo 5

Manual de Instalação

Neste breve capítulo, será descrito o processo de como utilizar as rotinas mencionadas no capítulo anterior. A implementação foi feita no Maple 12, mas também foi testada no Maple 7, desse modo, recomenda-se utilizar uma versão do Maple posterior à versão 7. Para fazer uso da rotinas, basta abrir o arquivo *conCur.mws* e compilar o código no Maple. Para fazer a compilação, basta executar todas elas (pressionando repetidamente a tecla *Enter* até o fim do arquivo, ou clicar no botão *Executetheworksheet*, cujo ícone é !!!). Uma vez compiladas as rotinas podem ser utilizadas como descrito no capítulo anterior.

Capítulo 6

Lista de Procedimentos

- (1) calcularDerivacaoFuncao 20
- (2) calcularDerivacao 20
- (3) calcular1FormaDerivacao 21
- (4) calcularConexao 22
- (5) calcularFuncional 23
- (6) acharRepresentanteFuncional 24
- (7) leviCivita 25
- (8) leviCivitaMatriz 26
- (9) comutador 27
- (10) calcularOperadorCurvatura 28
- (11) calcularProdutoExterior1Formas 29
- (12) calcularDerivadaExterior1Forma 30
- (11) calcularOperadorCurvaturaMatriz 31

Referências Bibliográficas

- [1] Teixeira, Renato, *Conexões e Curvaturas: Uma Abordagem Algébrica*, Dissertação de Mestrado DMAT-UFPE, (2009).
- [2] Do.Carmo, M., *Geometria Riemanniana*, IMPA, 1979.
- [3] Geddes, Keith O. & Czapor, S. R. & Labahn, G., *Algorithms for Computer Algebra.*, Boston: Kluwer Academic Publishers, 1992.
- [4] Lenimar Nunes de Andrade, *Introdução à Computação Algébrica com o Maple*, SBM, 2004.
- [5] Marcos Luiz Henrique, *Derivações e Campos de Vetores*, Dissertação de Mestrado DMAT-UFPE, (2001).
- [6] Michael Spivak, *Calculus on Manifolds* , (1965).
- [7] Jacqueline Rojas, *O Funcional de Yang-Mills*, Dissertação de Mestrado, DMAT-UFPE, (1997).