

Extensions des Algèbres de Lie

RANDRIAMAHERY Tojoniaina Mamitiana

01 mars 2025

Plan

- Introduction
- Quelques définitions et propositions
- Conclusion

Introduction :

La théorie des extensions de groupes et leur interprétation en termes de cohomologie est bien connue. Nous exposons le cas des algèbres de Lie (pour les extensions non abéliennes), avec un accent particulier sur les liens avec la théorie des dérivées extérieures covariantes, la courbure et l'identité de Bianchi en géométrie différentielle.

Définition 1 : Algèbre de Lie abélienne sur V

Soit V un espace vectoriel . On peut lui donner une structure d'algèbre de Lie abélienne en posant :

$$[u, v]_V = 0$$

$$, \forall u, v \in V$$

Définition 2 : Extension d'algèbres de Lie

Une extension de l'algèbre de Lie \mathfrak{g} par \mathfrak{B} est une algèbre de Lie $\tilde{\mathfrak{g}}$ telle qu'il existe une suite exacte courte :

$$0 \rightarrow V \rightarrow \tilde{\mathfrak{g}} \rightarrow \mathfrak{g} \rightarrow 0$$

Proposition 1 : Définition du crochet pour une extension

Supposons que l'algèbre de Lie \mathfrak{g} se représente sur V par $\eta : \mathfrak{g} \rightarrow \text{End}(V)$, et que $\alpha \in Z^2(\mathfrak{g}, V)$ soit un cocycle. Sur l'espace vectoriel $\tilde{\mathfrak{g}} = \mathfrak{g} \oplus V$, on peut définir un crochet de Lie, pour tous $x + v, y + w \in \tilde{\mathfrak{g}}$ par :

$$[x + v, y + w] = [x, y]_{\mathfrak{g}} + \eta(x)w - \eta(y)v + \alpha(x, y)$$

Preuve :

Montrons que ce crochet satisfait l'identité de Jacobi. Soit $(x + v), (y + w), (z + t) \in \tilde{\mathfrak{g}}$. On doit vérifier que :

$$[x + v, [y + w, z + t]] + [y + w, [z + t, x + v]] + [z + t, [x + v, y + w]] = 0.$$

Calcul :

$$\begin{aligned} [x + v, [y + w, z + t]] &= [x + v, [y, z] + \eta(y)t - \eta(z)w + \alpha(y, z)] \\ &= [x, [y, z]] + \eta(x)(\eta(y)t - \eta(z)w + \alpha(y, z)) \\ &\quad - \eta([y, z])v + \alpha(x, [y, z]). \end{aligned}$$

Répétant ce calcul pour les deux autres termes, puis utilisant $d\alpha = 0$ et les propriétés de η , on montre que la somme cyclique est nulle. \square

Proposition 2 : Modification du cocycle par un cobord

Si l'on remplace α par un cocycle cohomologique équivalent $\alpha + d\beta$, avec $\beta \in C^1(\mathfrak{g}, V)$, le crochet dévient :

$$[x + v, y + w]_{\beta} = [x, y]_{\mathfrak{g}} + \eta(x)w - \eta(y)v + \alpha(x, y) + d\beta(x, y)$$

où :

$$d\beta(x, y) = \eta(x)\beta(y) - \eta(y)\beta(x) - \beta([x, y])$$

Preuve :

En remplaçant α par $\alpha + d\beta$ dans la définition du crochet, on a directement :

$$[x + v, y + w]_{\beta} = [x, y]_{\mathfrak{g}} + \eta(x)w - \eta(y)v + \alpha(x, y) + d\beta(x, y).$$

Le terme supplémentaire $d\beta(x, y)$ s'obtient par la définition du cobord $d\beta$, ce qui conclut la preuve. \square

Proposition 3 : Isomorphisme des structures d'algèbres de Lie

Considérons l'isomorphisme d'espaces vectoriels

$$\phi : \tilde{\mathfrak{g}} \rightarrow \tilde{\mathfrak{g}}$$

, défini par :

$$\phi(x + v) = x - \beta(x) + v.$$

Cet isomorphisme vérifie :

$$\phi([x + v, y + w]) = [\phi(x + v), \phi(y + w)]_{\beta}.$$

Cela montre que les deux crochets, celui défini par α et celui défini par $\alpha + d\beta$, conduisent à des structures d'algèbres de Lie isomorphes .

Preuve :

En appliquant ϕ , on obtient :

$$\begin{aligned}\phi([x + v, y + w]) &= \phi([x, y] + \eta(x)w - \eta(y)v + \alpha(x, y)) \\ &= [x, y] - \beta([x, y]) + \eta(x)w - \eta(y)v + \alpha(x, y).\end{aligned}$$

De l'autre côté, $[\phi(x + v), \phi(y + w)]_\beta$ donne le même résultat, car β se simplifie dans les termes crochets et les actions. Cela conclut la preuve. \square

Conclusion :

Dans ce travail, nous avons étudié les extensions d'algèbres de Lie, en nous concernant sur leur structure et leurs propriétés algébriques. Nous avons montré que toute extension d'une algèbre de Lie donnée peut être décrite à l'aide d'un cocycle dans la cohomologie de Chevalley-Eilenberg, mettant ainsi en évidence le rôle central de la théorie des cohomologies dans la classification des extensions.

MERCI