Suite récurente non-linéaire

Problème avec correction proposé par Stéphan Tassart
IRCAM, 1 place Igor Stravinsky, 75004 PARIS

`tassart@ircam.fr` `http://www-stud.enst.fr/~tassart/`

Décembre 1996

Résumé

Les fonctions de substitution sont parmi les plus utiles pour manipuler des expressions par l'intermédiaire d'un outil de calcul formel. L'exemple suivant illustre une utilisation fréquente de la fonction `subs` de maple appliqué au cas de la détermination du terme général d'une suite récurente non-linéaire.

1 Suite récurente non linéaire

Le problème consiste à identifier les solutions polynômiales de la suite définie comme suit:

$$\forall n \in \mathbb{N} \quad n u_{n+2} - 5 u_{n+1} - (n+1) u_n = 0 \tag{1}$$

- On suppose que u_n est un polynôme U de degré d: $U = \sum_{k=0}^{d} a_k n^k$. On se fixe, pour faire les calculs sous Maple, un degré d assez grand. Définir le polynôme U `[sum]`.
- Montrer que la fonction `[subs]` permet d'obtenir les expressions de u_{n+2} puis de u_{n+1}.
- Définir le polynôme $Q = n u_{n+2} - 5 u_{n+1} - (n+1) u_n$. Développer ce polynôme `[expand]`, puis regrouper les termes selon les puissances de n `[collect]`.
- Le polynôme U est une solution du problème si et seulement si le polynôme Q est identiquement nul, c'est-à-dire si tous ses coefficients sont nuls. Définir sous Maple l'ensemble des équations que doivent vérifier les coefficients a_k pour que U soit effectivement une solution. `[coeffs]`.
- Résoudre le système de $d+1$ équations à $d+1$ inconnues `[solve]` (rem: pour définir l'ensemble des inconnues, vous pourrez utiliser la syntaxe suivante: `{a[k]$k=0..d}`).
- Reporter les solutions a_k dans U `[subs]`.
- Factoriser la solution `[factor]`.
- Conclure.

2 Solutions

Pour l'instant nous définissons un polynôme de degré d assez grand (mais pas trop grand quand même!):

```
> d:=10;
> U:=sum(a[k]*n^k,k=0..d);
```

Le polynôme Q se construit facilement à partir de U par l'intermédiaire de l'instruction suivante:

```
> Q:= n*subs(n=n+2,U) - 5*subs(n=n+1,U) + (n+1)*U;
```

On développe, puis on regroupe les termes selon les puissances de n:

```
> expand(");
> collect(",n);
```

Le système d'équation que doivent vérifier les coefficients a_k est alors le système suivant:

```
> equ_set := {coeffs(",n)};
```

Il ne reste plus alors qu'à résoudre le système:

```
> solve(equ_set, {a[k]$k=0..d});
```

puis à reporter les solutions dans U:

```
> subs(",U);
```

et à factoriser:

```
> factor(");
```

En conclusion, les seules solutions polynomiales qui conviennent pour u_n, sont proportionnelles au polynôme $n(n-1)(2n-1)$.