Programación: métodos de Runge–Kutta implícitos para resolver EDO's con valores iniciales

Objetivos. Programar algunos métodos de Runge–Kutte implícitos para resolver ecuaciones diferenciales ordinarias con condiciones iniciales. Hacer comprobaciones y comparar la eficiencia con otros métodos.

Requisitos. Haber programado y probado el método de Euler y algunos métodos de Runge–Kutta explícitos.

1. Ejemplos y algoritmos de las clases anteriores. En las clases anteriores ya hemos programado un ejemplo `f1` de función del lado derecho de ecuación diferenciales, un ejemplo `x1` de solución correspondiente, el esquema general `onestepmethod` de métodos de un paso, y algunas funciones que hacen la comprobación. En esta clase se propone usar todas estas funciones.

2. Un paso del método implícito de Heun (o Runge–Kutta de segundo orden).

```
defun heunimplstep(f, t, v, h):
   tol <- 1.0e-14
   er <- tol + 1
   k1 <- h * f(t, v)
   vnext <- v + k1
   while er > tol:
      vsave <- vnext
      k2 <- h * f(t + h, vnext)
      vnext <- v + (k1 + k2) / 2
      er <- abs(vnext - vsave)
   vnext
```

Se propone probar este método `heunimplstep` en vez de `eulerstep` u otras esquemas (`heunstep`, `rk41step`, etc.). Hacer pruebas, observar el error máximo y comparar el tiempo de ejecución con otros métodos.

3. Fórmula matemática para este método implícito. Para entender el sentido matemático del algoritmo programado se recomienda escribir la fórmula recursiva (sustituir las fórmulas para `k1` y `k2`):

$$v^{[s+1]} = v + h\,\frac{f(t,v) + f(t+h, v^{[s]})}{2}. \tag{1}$$

Suponiendo que la sucesión $(v^{[s]})_{s=0}^{\infty}$ tiene un límite w y la función f es continua, en ambos lados de la igualdad (1) pasar al límite cuando s tiende a infinito:

$$w = \tag{2}$$

La fórmula (2) es la propiedad que satisface el punto límite de las iteraciones. Es el valor que aproximamos en un paso del algoritmo de Heun implícito. Nótese que w participa en ambos lados de la ecuación, por eso la fórmula se llama *implícita*.

4. El método de Euler implícito. El valor nuevo w debe satisfacer la ecuación

$$w = v + hf(t+h, w).$$

Busquemos w con el método de iteración simple, como el límite de una sucesión $(v^{[s]})_{s=0}^{\infty}$. Pongamos

$$v^{[0]} = v + hf(t+h, v)$$

y

$$v^{[s+1]} = v + hf(t+h, ???).$$

Escribimos el algoritmo correspondiente:

```
defun eulerimplstep(f, t, v, h):
   tol <- 1.0e-14; er <- tol + 1
   vnext <- v + h * f(t + h, v)
   while er > tol:
      vsave <- vnext
      vnext <- v + h * f(???, ???)
      er <- abs(vnext - vsave)
   vnext
```

Haga pruebas para este método.