3. Intégration numérique

3.4 Exercices

Exercice 15

```
> restart;
> intrap := proc( f, a, b, N )
  local h, int :
  h := (b - a)/N :
  int := h·( subs( x = a, f ) + subs( x = b, f ) )/2 :
  int := int + h·add( subs( x = a + k·h, f ), k = 1 ..N - 1 ) :
  end proc:
> intrap( x, 0, 1, 10 );
```

$$\frac{1}{2} \qquad \textbf{(1.1.1.1.1)}$$

```
>
```

Exercice 16

```
> intsimp := proc( f, a, b, N )
  local h, int :
  h := (b - a)/N :
  int := add( (h/3)·subs( x = a + 2·k·h, f ), k = 0 .. N/2 - 1 ) :
  int := int + add( (4·h/3)·subs( x = a + (2·k + 1)·h, f ), k = 0 .. N/2 - 1 ) :
  int := int + add( (h/3)·subs( x = a + (2·k + 2)·h, f ), k = 0 .. N/2 - 1 ) :
  end proc:
> intsimp( x, 0, 1, 10 );
```

$$\frac{1}{2} \qquad \textbf{(1.1.1.2.1)}$$

```
> intsimp( x^2, 0, 1, 10 );
```

$$\frac{1}{3} \qquad \textbf{(1.1.1.2.2)}$$

```
> intsimp( x^3, 0, 1, 10 );
```

(1.1.1.2.3)

$$\frac{1}{4} \qquad \textbf{(1.1.1.2.3)}$$

```
> evalf(intsimp(x^4, 0, 1, 10));
```

$$0.2000133333 \qquad \textbf{(1.1.1.2.4)}$$

```
>
```

Exercice 17

```
> intrecd := proc(f, a, b, N)
  local h :
  h := (b - a)/N :
  add(h*subs(x = a + (k + 1)*h, f), k = 0 ..N - 1) :
  end proc:
> intrecd(x, 0, 1, 10);
```

$$\frac{11}{20} \qquad \textbf{(1.1.1.3.1)}$$

```
> evalf(intrecd(x, 0, 1, 10000));
```

$$0.5000500000 \qquad \textbf{(1.1.1.3.2)}$$

```
>
>
>
```

Exercice 20

```
> L := [0, 10.5, 35, 45, 40.5, 25, 20.5, 29, 27, 12.5, 0] :
> f := x->x^2 :
  L2 := map(f, L) :
> N := 10 : a := 0 : b := 0.1 : h := (b - a)/N :
> intV2 := add(h/3*L2[2*k + 1] + 4*h/3*L2[2*k + 1 + 1] +
  h/3*L2[2*k + 2 + 1], k = 0 .. N/2 - 1) :
> Vrms := sqrt(1/(b - a)*intV2);
```

$$Vrms := 27.72423729 \qquad \textbf{(1.1.1.4.1)}$$

```
>
```

5. Equations différentielles

```
> euler2 := proc( f, y0, t0, T, N )
  local y, h, n :
  y := array(0..N) :
  y[0] := y0 :
  h := (T - t0)/N :
  for n from 0 to N - 1 do
  y[n + 1] := y[n] + h·f(t0 + n·h, y[n]) :
  od:
  y :
  end proc:
>
> runge := proc( f, y0, t0, T, N )
  local y, h, n, k1, k2 :
  y := array(0..N) :
  y[0] := y0 :
  h := (T - t0)/N :
  for n from 0 to N - 1 do
  k1 := f(t0 + n·h, y[n]);
  k2 := f(t0 + n·h + h/2, y[n] + h/2·k1) :
  y[n + 1] := y[n] + h·k2 :
  od:
  y :
  end proc:
>
> rk4 := proc( f, y0, t0, T, N )
  local y, h, n, k1, k2, k3, k4 :
  y := array(0..N) :
  y[0] := y0 :
  h := (T - t0)/N :
  for n from 0 to N - 1 do
  k1 := f(t0 + n·h, y[n]);
  k2 := f(t0 + n·h + h/2, y[n] + h/2·k1) :
  k3 := f(t0 + n·h + h/2, y[n] + h/2·k2) :
  k4 := f(t0 + (n + 1)·h, y[n] + h·k3) :

  y[n + 1] := y[n] + h·(k1 + 2·k2 + 2·k3 + 4·k4)/6 :
  od:
  y :
  end proc:
```