Fixed-point iteration approximations

Inverse of a number

```
> restart;
g1:=x->2*x-A*x^2;
```

$$g1 := x \rightarrow 2\,x - A\,x^2 \tag{1}$$

```
> A:=7;
```

$$A := 7 \tag{2}$$

```
> p0:=0.2;
```

$$p0 := 0.2 \tag{3}$$

```
> for n from 1 to 20 do
  p[n]:=g1(p0);
  err:=abs(p[n]-p0);
  if err>=10^(-8) then
  p0:=p[n];
  else
  break
  end if
  end do;
```

$$\begin{gathered} p_1 := 0.12 \\ err := 0.08 \\ p_2 := 0.1392 \\ err := 0.0192 \\ p_3 := 0.14276352 \\ err := 0.00356352 \\ p_4 := 0.1428570815 \\ err := 0.0000935615 \\ p_5 := 0.1428571429 \\ err := 6.14\ 10^{-8} \\ p_6 := 0.1428571428 \\ err := 1.\ 10^{-10} \end{gathered} \tag{4}$$

```
> evalf(1/7);
```

$$0.1428571429 \tag{5}$$

square root

```
> restart;
> g2:=x->(1/2)*x+a/(2*x);
```

$$g2 := x \rightarrow \frac{1}{2}\,x + \frac{1}{2}\,\frac{a}{x} \tag{6}$$

```
> a:=6;
```

$$a := 6 \tag{7}$$

```
> p0:=2.1;
```

$$p0 := 2.1 \tag{8}$$

```
> for n from 1 to 20 do
  p[n]:=g2(p0);
  err:=abs(p[n]-p0);
  if err>=10^(-8) then
  p0:=p[n];
  else
  break
  end if
  end do;
```

$$\begin{gathered} p_1 := 2.478571428 \\ err := 0.378571428 \\ p_2 := 2.449660354 \\ err := 0.028911074 \\ p_3 := 2.449489749 \\ err := 0.000170605 \\ p_4 := 2.449489742 \\ err := 7.\,10^{-9} \end{gathered} \tag{9}$$

```
> evalf(sqrt(6));
```

$$2.449489743 \tag{10}$$