
```
1: for  $i = \{1, \dots, n\}$  do
2:    $\mathcal{B} \leftarrow \text{SUBDIVIDE}(\mathcal{B})$ 
3:    $\mathcal{B} \leftarrow \mathcal{B} \cap f(\mathcal{B})$ 
4: return  $\mathcal{B}$ 
```

```

1 function relative_attractor(t, f, X, steps)
2 dim = t.dim; hit = 1; sd = 8; tic;
3 for s = 1:steps,
4     t.set_flags('all', sd);
5     t.subdivide(sd);
6     b = t.bboxes(-1); N = size(b,2);
7     S = whos('X'); l = floor(5e7/S.bytes);
8     for k = 0:floor(N/l),
9         K = k*l+1:min((k+1)*l,N);
10        c = b(1:dim,K);
11        r = b(dim+1:2*dim,1);
12        n = size(c,2); E = ones(n,1);
13        P = kron(E,X)*diag(r) + ...
14            kron(c',ones(size(X,1),1));
15        t.set_flags(f(P)', hit);
16    end
17    t.remove(hit);
18    fprintf(...
19        'depth %d, %d boxes, %.1f sec\n',...
20        t.depth,t.count(-1),toc...
21    );
22 end

```

```
1  function relative_attractor(f::BoxMap, B::BoxSet)
2      for s in 1:steps
3          B = subdivide(B)
4          B = B ∩ f(B)
5      end
6      return B
7  end
```