1: **for** $i = \{1, \ldots, n\}$ **do**
2: $\quad \mathcal{B} \leftarrow \textsc{subdivide}(\mathcal{B})$
3: $\quad \mathcal{B} \leftarrow \mathcal{B} \cap f(\mathcal{B})$
4: **return** $\mathcal{B}$

```
function relative_attractor(t, f, X, steps)
dim = t.dim; hit = 1; sd = 8; tic;
for s = 1:steps,
    t.set_flags('all', sd);
    t.subdivide(sd);
    b = t.boxes(-1); N = size(b,2);
    S = whos('X'); l = floor(5e7/S.bytes);
    for k = 0:floor(N/l),
        K = k*l+1:min((k+1)*l,N);
        c = b(1:dim,K);
        r = b(dim+1:2*dim,1);
        n = size(c,2); E = ones(n,1);
        P = kron(E,X)*diag(r) + ...
            kron(c',ones(size(X,1),1));
        t.set_flags(f(P)', hit);
    end
    t.remove(hit);
    fprintf(...
        'depth %d, %d boxes, %.1f sec\n',...
        t.depth,t.count(-1),toc...
    );
end
```

```
    function relative_attractor(f::BoxMap, B::BoxSet)
        for s in 1:steps
            B = subdivide(B)
            B = B ∩ f(B)
        end
        return B
    end
```