Parareal implementation

Consider the following ODE:

$$\begin{cases} \dfrac{du}{dt} = f(t,u)\,, \quad t \in [t_0, t_N] \\ u(t_0) = u_0 \end{cases}$$

Introduce the following propagation operator:

- **Coarse propagator $\mathcal{G}$** (e.g. Forward Euler with $h \sim dT$)
- **Fine propagator $\mathcal{F}$** (e.g. Forward Euler with $dt = dT/50$)

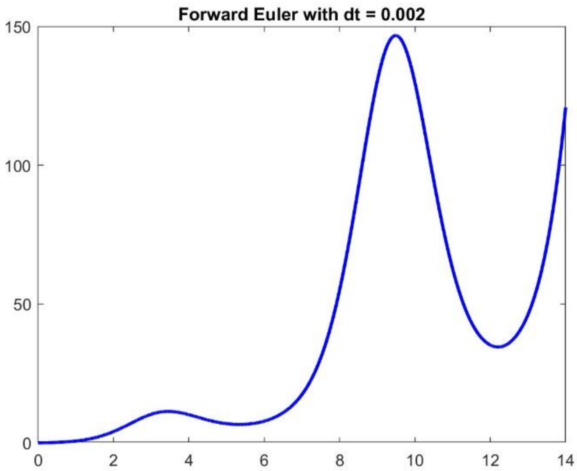
Notation: $U_n^k \approx u(t_n)$ at the step k, with $t_n = t_0 + n * dT$

My parareal

$$\begin{cases} u' = \mathrm{u} * \sin(t) + t\,, \quad t \in [t_0, t_N] \\ u(t_0) = 0 \end{cases}$$

Step ~ 0

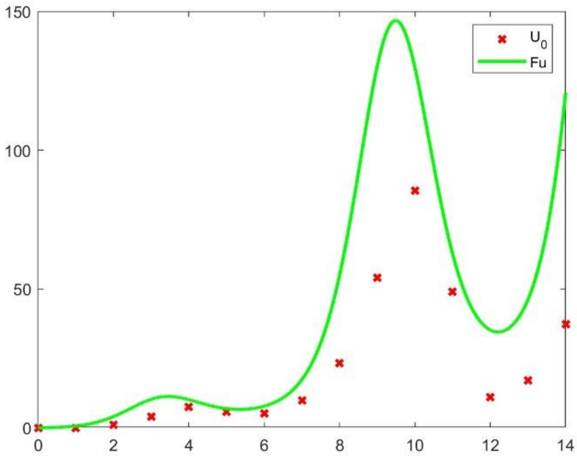
```
f = @(t,y) sin(t).*y + t;
t0 = 0; tN = 14;
y0 = 0;
dt = 0.002;

[t,y_fine] = fwd_Euler(t0,tN,y0,dt,f);
```



Step 1

Initial guess, first iteration of $\mathcal{G}$

```
dT = 1;
[t_coarse,U_0] = fwd_Euler(t0,tN,u0,dT,f);
```



Definitions of u_fine and t_fine

```
for m = 1 : (tN - t0) / dT % from 1 to number of coarse subintervals
    t_fine(m,:) = [t0 + (m-1) * dT : dt : t0 + m * dT]; % fine approx. of each subint.
end
% Notice: t_fine(m,end) = t_fine(m+1,1)

% L_coarse = n of coarse subintervals, L_fine = n of fine subinterval
u_fine = zeros(L_coarse, L_fine);

U = U_0;
```

Iterative Step

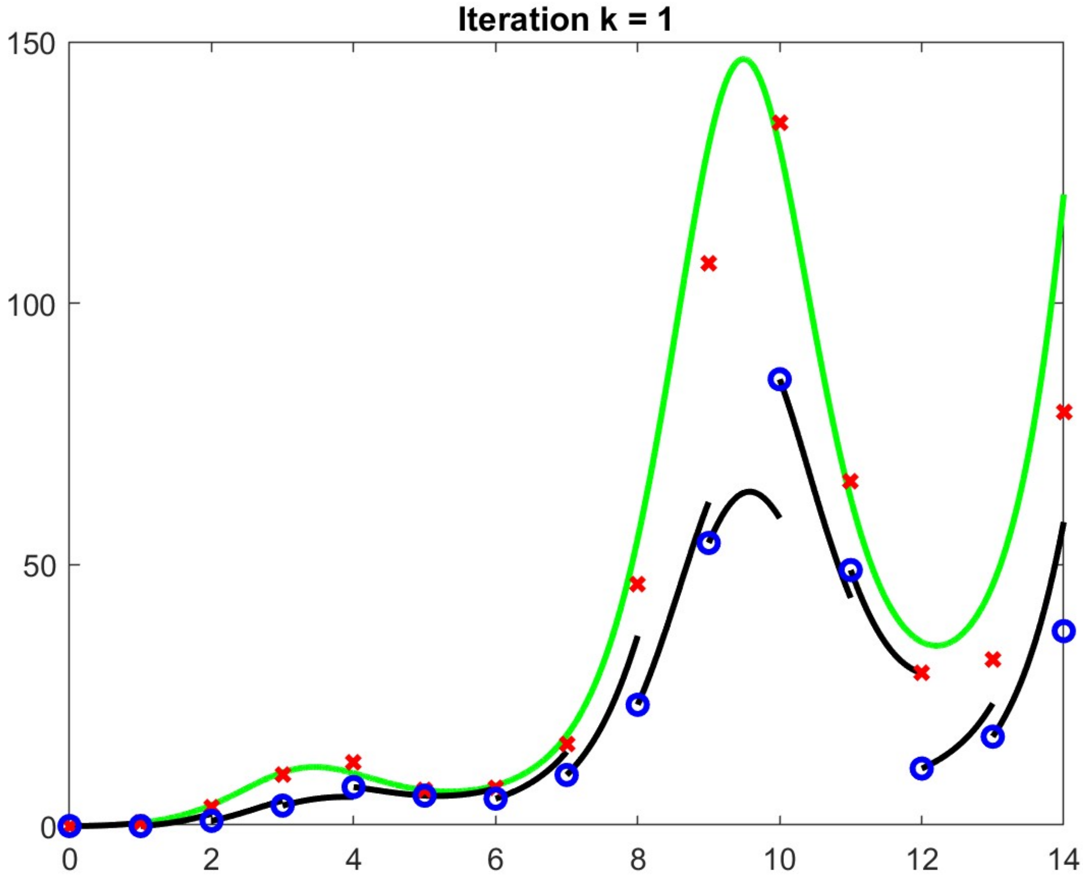
```
for k = 1 : k_max

    % Evaluation of $\mathcal{F}(t_{n+1}, t_n, U_n^k)$
    % Parallel step: computation of the fine solution on each subinterval
    parfor n = 1 : L_coarse - 1
        [t_fine(n,:),u_fine(n,:)] = fwd_Euler(t_coarse(n),t_coarse(n+1),U_0(n),dt,f);
    end
```

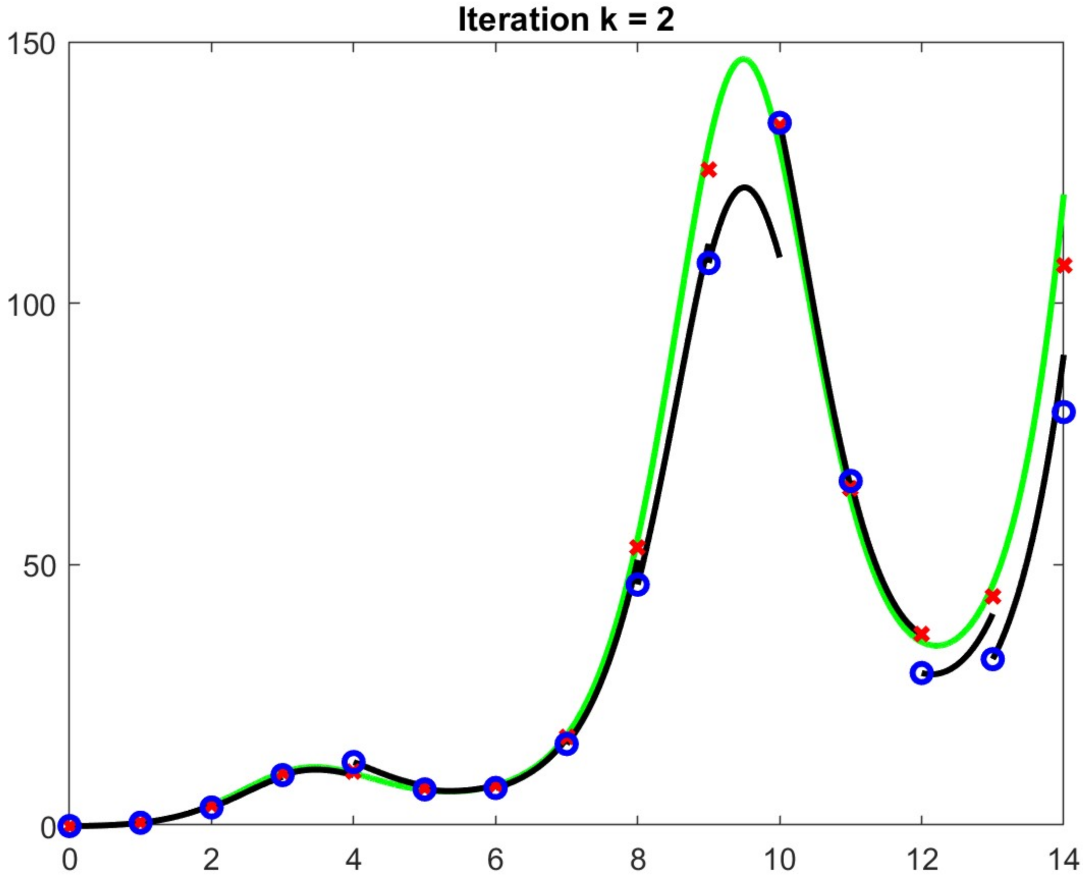
$\mathcal{F}$

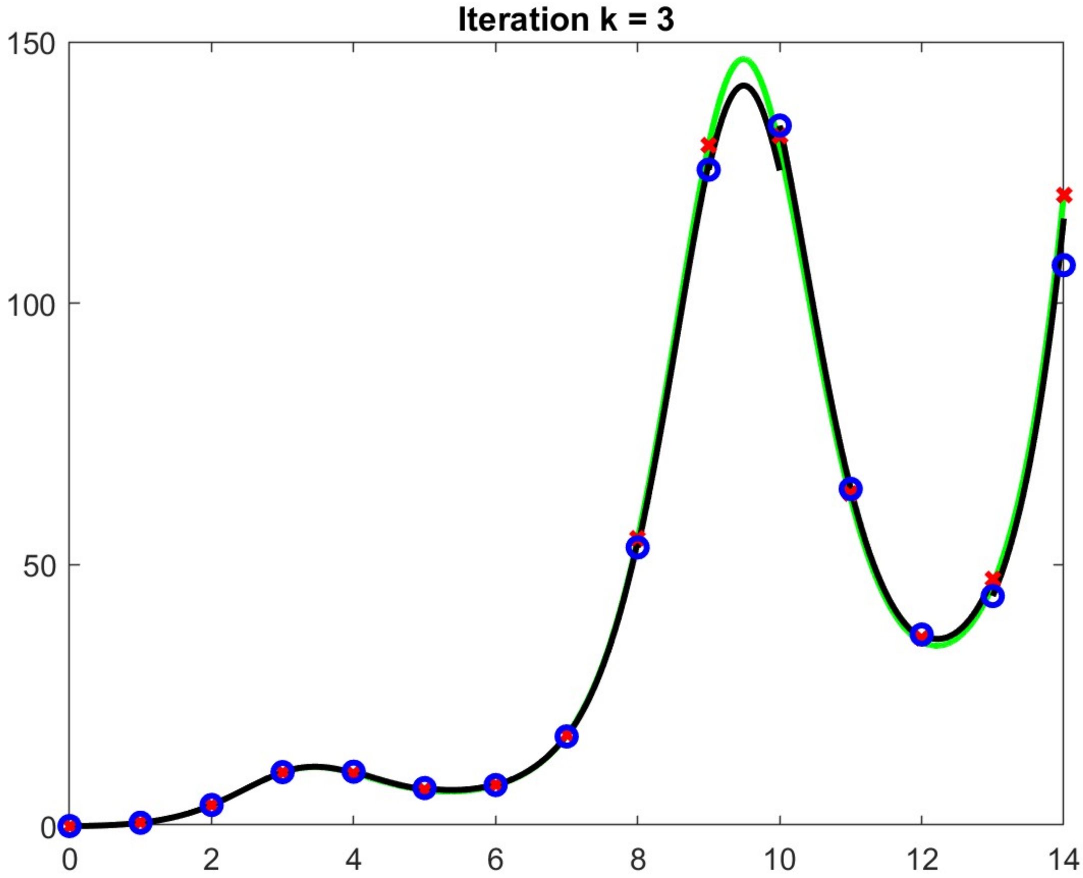
Between two successive coarse time steps

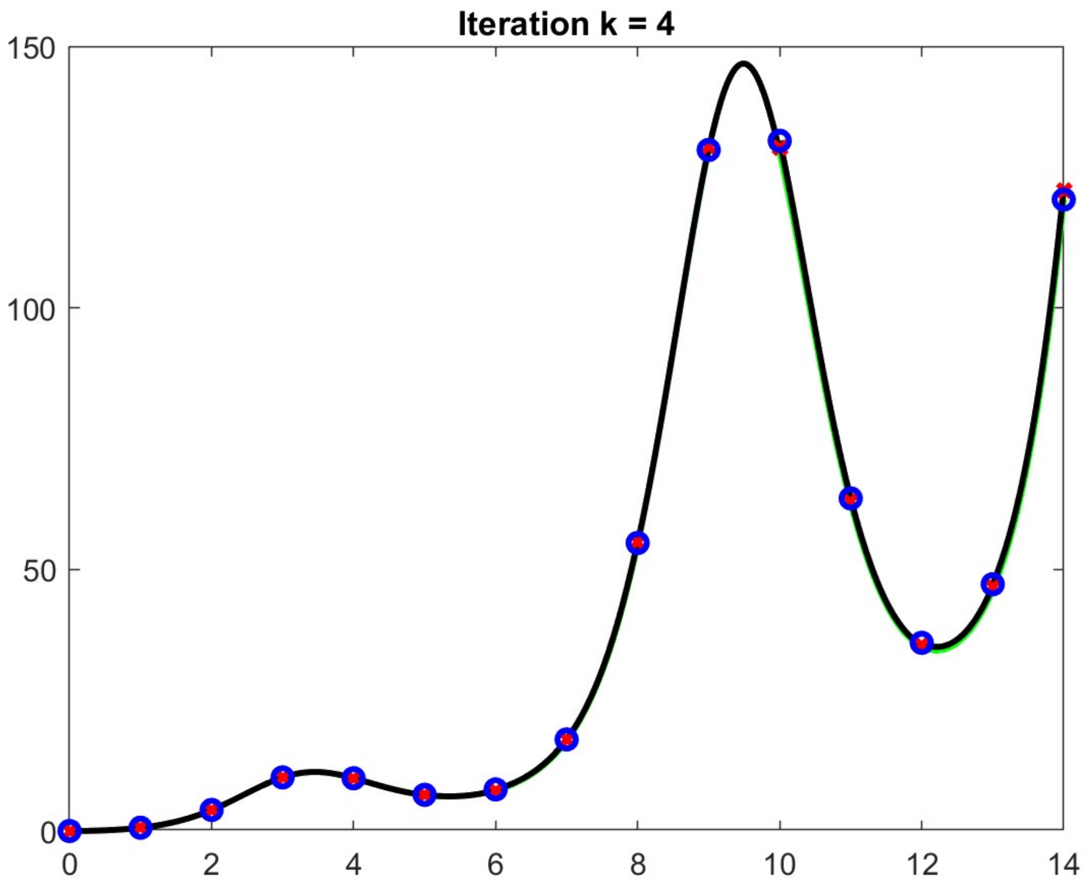
```
    % Evaluation of $\mathcal{G}(t_{n+1}, t_n, U_n^k)$
    % Update of the coarse solution
    for h = 1 : L_coarse - 1
        du = sin(t_coarse(h))*U(h) + t_coarse(h);
        U_k(h+1) = U(h) + dT*du;
    end


    % Evaluation of $\mathcal{G}(t_{n+1}, t_n, U_n^{k+1})$
    for n = 1 : L_coarse - 1
        du = sin(t_coarse(n))*U(n) + t_coarse(n);
        U(n+1) = U(n) + dT*du;

        % Correction step $U_{n+1}^{k+1} = \mathcal{G}(t_{n+1}, t_n, U_n^{k+1}) + \mathcal{F}(t_{n+1}, t_n, U_n^k) - \mathcal{G}(t_{n+1}, t_n, U_n^k)$

        U(n+1) = U(n+1) + u_fine(n, end) - U_k(n+1);
    end

    U_0 = U

End
```

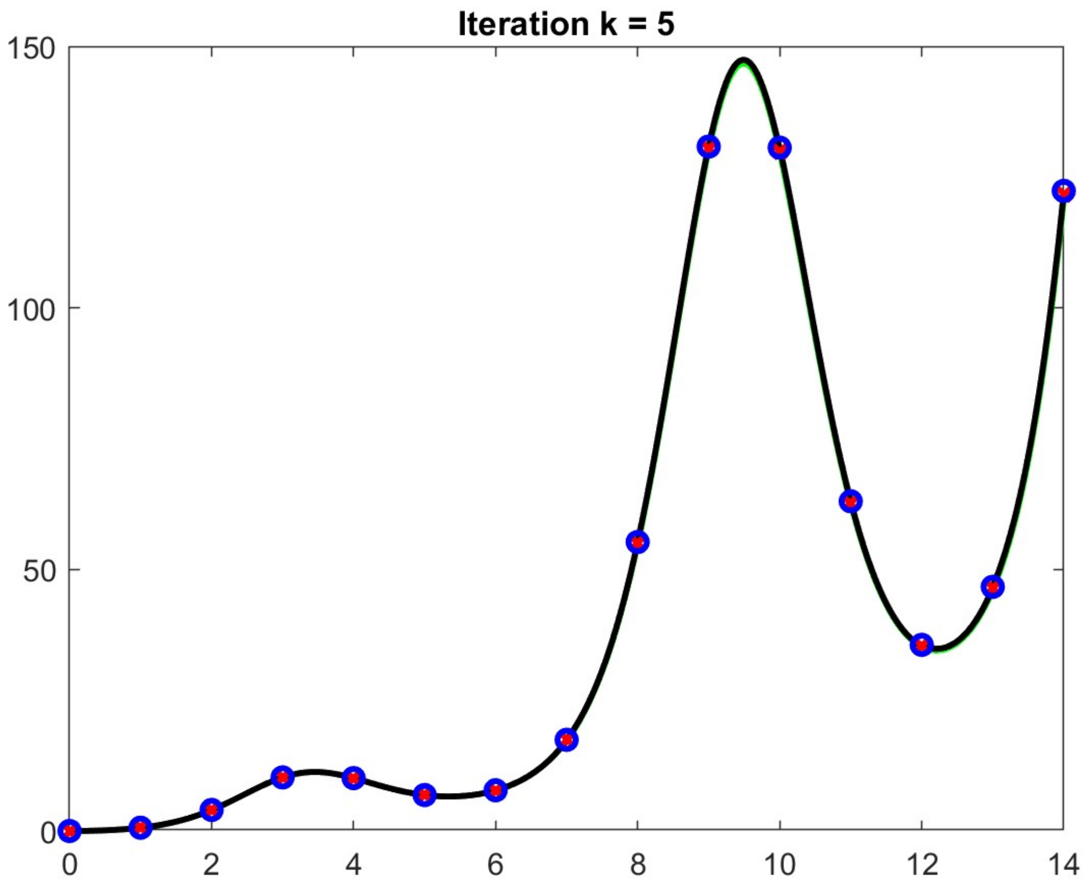
..Some plots now..

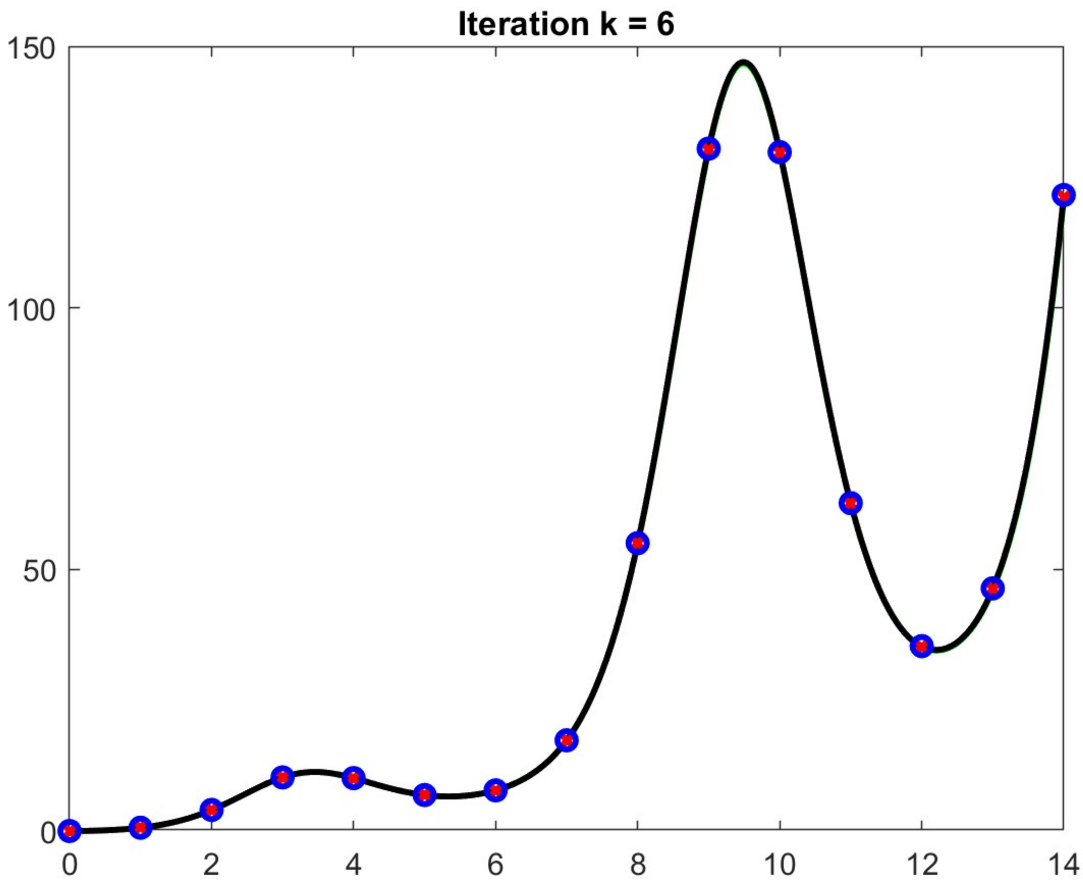

Iteration k = 1
150
100
50
0
0
2
4
6
8
10
12
14

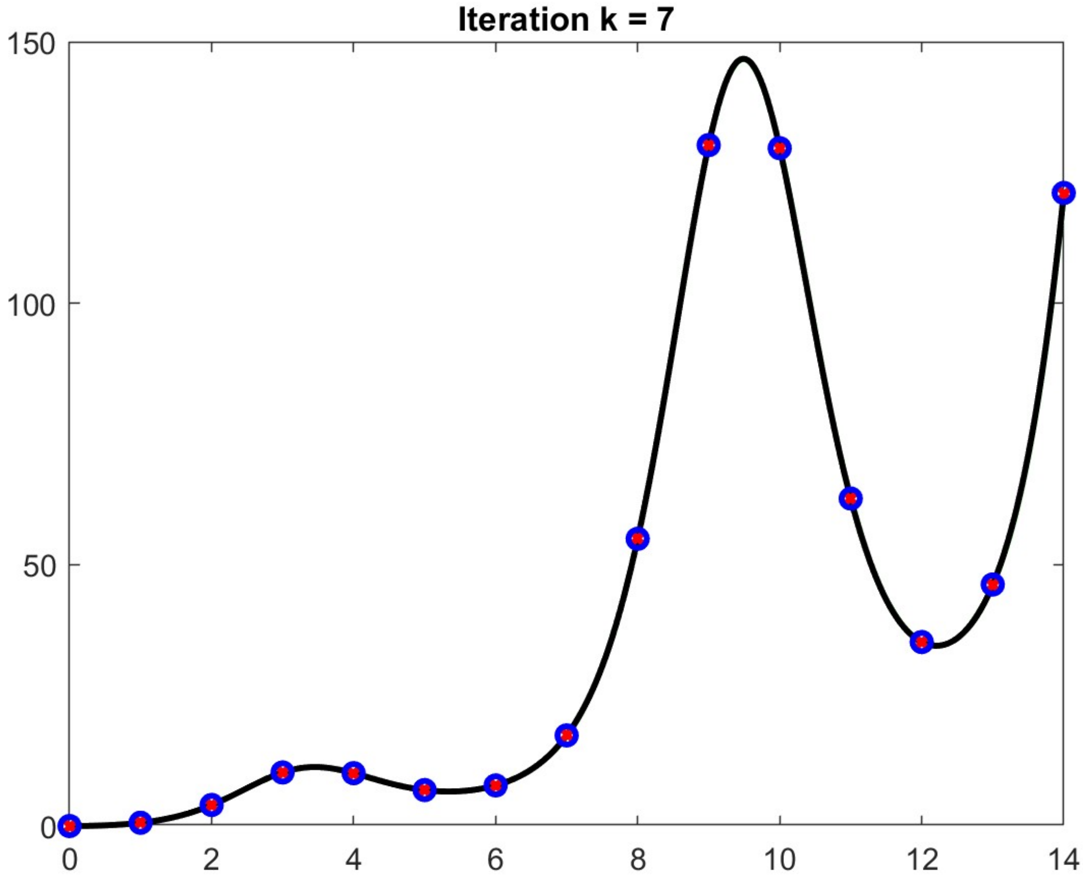

Iteration k = 2
150
100
50
0
0
2
4
6
8
10
12
14

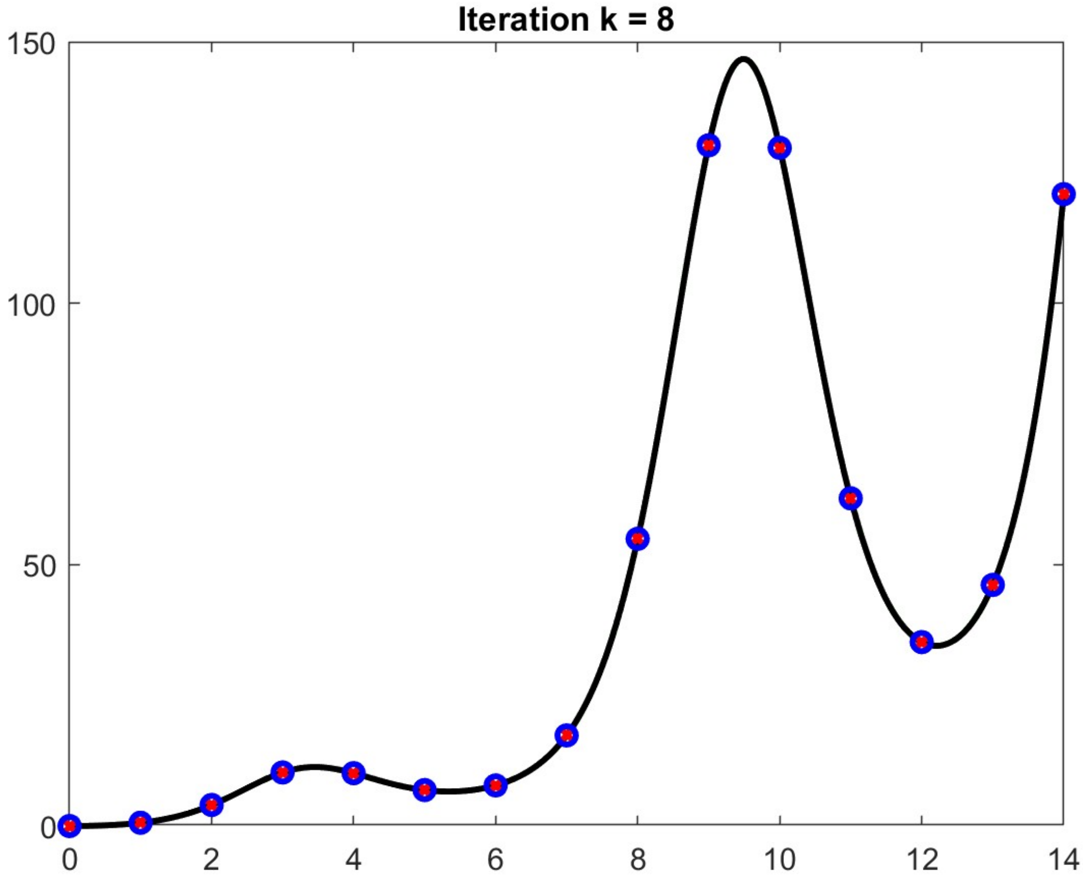

Iteration k = 3
150
100
50
0
0
2
4
6
8
10
12
14


Iteration k = 4
150
100
50
0
0
2
4
6
8
10
12
14


Iteration k = 5
150
100
50
0
0
2
4
6
8
10
12
14


Iteration k = 6
150
100
50
0
0
2
4
6
8
10
12
14


Iteration k = 7
150
100
50
0
0
2
4
6
8
10
12
14


Iteration k = 8
150
100
50
0
0
2
4
6
8
10
12
14