8.5.3. Question:

[3 + 3 points]

Make a guess of how many letters the network can store. Then create a (small) set of letters. Check if all letters of your list are fixed points under the network dynamics.

```python
In [56]:
# write your code here
# write your code here
# the letters we want to store in the hopfield network
letter_list = ['A', 'B', 'C', 'D', 'E', 'F', 'G']

abc_dictionary = pattern_tools.load_alphabet()

print("The alphabet is stored in an object of type: {}",format(type(abc_dictionary)))

# access the first element and get it's size (they are all of same size)
pattern_shape = abc_dictionary[\'A\'].shape

print("Letters are patterns of size: ",format(pattern_shape))

# create an instance of the class HopfieldNetwork
hopfield_net = network.HopfieldNetwork(nr_neurons=pattern_shape[0]*pattern_shape[1])

# create a list using Python's List Comprehension syntax:
pattern_list = [abc_dictionary[key] for key in letter_list]
plot_tools.plot_pattern_list(pattern_list)

# store the patterns
hopfield_net.store_patterns(pattern_list)

for i, pattern in enumerate(pattern_list):
    hopfield_net.set_state_from_pattern(pattern)

# from this initial state, let the network dynamics evolve.
states = hopfield_net.run_with_monitoring(nr_steps=4)

# each network state is a vector, reshape it to the same shape used to create the patterns.
states_as_patterns = pattern_tools.reshape_patterns(states, pattern_list[0].shape)

# plot the states of the network
plot_tools.plot_state_sequence_and_overlap(
    states_as_patterns, pattern_list, reference_idx=0, suptitle="Network dynamics")
```

The alphabet is stored in an object of type: <type 'dict'>

letters are patterns of size: (100, 100). Create a network of corresponding size.

![Network dynamics](image1)

![Network dynamics](image2)

![Network dynamics](image3)