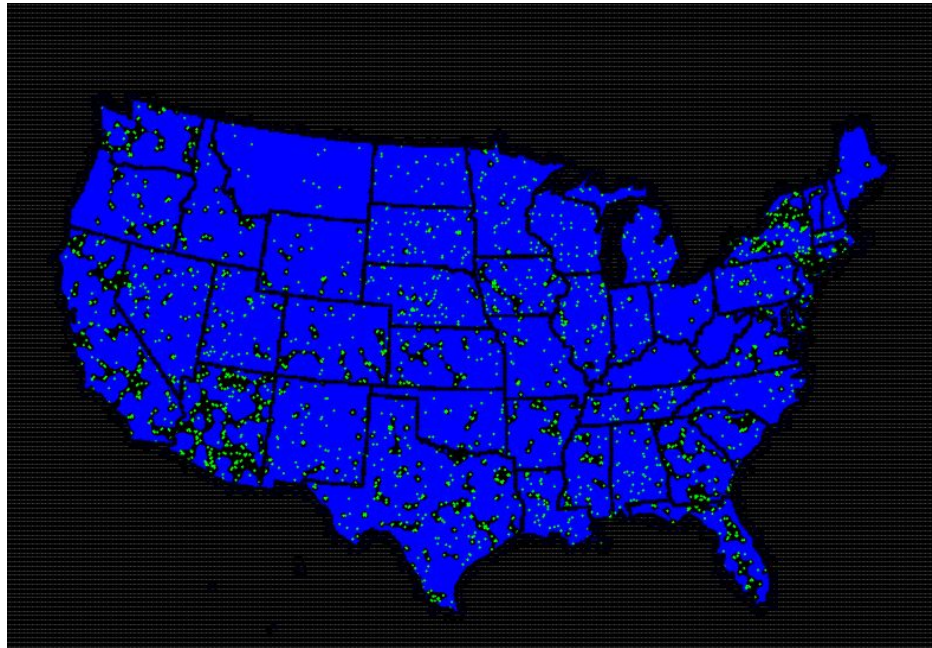


Symulacja rozwoju epidemii SARS-nCoV-2 w Stanach Zjednoczonych



Kamil Serafin

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej

Automatyka i Robotyka, III rok

Tworzenie aplikacji do symulacji i sterowania procesów dyskretnych

1. Wstęp

Moim zadaniem było opracowanie złożonej symulacji rozwoju epidemii choroby COVID-19 na terenie Stanów Zjednoczonych Ameryki. Zdecydowałem się na wybór tego państwa jako obiektu badań nie tylko z powodu najliczniejszych i najlepiej opracowanych danych statystycznych związanych z rozwojem pandemii na jego terenie w pierwszej połowie 2020 roku, ale również z powodu jego złożonej struktury, czyli podziału na 50 odrębnych stanów. Podział ten chciałem odpowiednio odwzorować w mojej symulacji, tak, by efekty działania programu były jak najbardziej zbliżone do rzeczywistości, a ostateczny wynik pokazywał przewidywany rozwój epidemii w sposób czytelny i atrakcyjny dla oka.

W odróżnieniu od innych studentów, podczas pracy miałem się skupić nie tylko na opracowaniu symulacji zależnej od liczby przypadków wystąpienia koronawirusa w kraju, ale również przedstawić jak rozwój epidemii w poszczególnych stanach wpływa na inne, sąsiadujące z nimi. Wymagało to wielu dni żmudnej pracy, związanej z odpowiednim opracowaniem mapy, podziałem administracyjnym i przeprowadzeniem badań dla każdego z 48 stanów USA (pod uwagę nie brano Alaski oraz Hawajów, jako leżących z dala od innych skażonych regionów). Wyniki owej pracy prezentuję w niniejszym opracowaniu, w którym przedstawię poszczególne etapy działania programu, omawiając przy tym jego najważniejsze założenia, wady i zalety.

Omawiany tutaj kod, zamieszczony został jako załącznik do niniejszego sprawozdania i jest jego integralną częścią. Program przesłany został w dwóch wersjach: samodzielnej, bez GUI, oraz zaktualizowanej do aplikacji okienkowej tworzonej w zespole o składzie: Marta Sosenkiewicz, Kamil Serafin, Aleksander Siwek, Karol Stryczek. Następujące w następnym paragrafie omówienie symulacji zostało przeprowadzone na wersji bez GUI (ów wariant zostanie omówiony na końcu) jako na szybszej i sprawiającej mniej problemów związanych z uruchomieniem.

2. Symulacja rozwoju epidemii krok po kroku

Po uruchomieniu symulacji poprzez realizację pliku *main.m*, program przystępuje do wykonania kolejnych kroków, które teraz przedstawię.

Generowanie plików wejściowych - Python

W przeciwieństwie do innych osób realizujących rozwój epidemii jako ostatnie ćwiczenie, miałem możliwość podejść do kwestii gromadzenia danych w sposób zgoła inny. Zdecydowałem się więc na niekorzystanie z arkuszy kalkulacyjnych, zamiast tego generując pliki txt, z odpowiednio wygenerowanymi stanami, z których korzysta następnie program w Matlabie.

Rozpoczynamy od wywołania odpowiedniej komendy w środowisku Matlaba (dzięki temu nie ma potrzeby wykonywania programu w Pythonie w osobnym środowisku):

```
system('python Python_workspace/generate_files.py');
```

Owe polecenie przystępuje do natychmiastowej realizacji pliku *generate_files.py* umieszczonego w podfolderze *Python_workspace*, poprzez wykonanie odpowiedniej komendy (podanej jako argument funkcji *system*) w systemie operacyjnym.

W tym miejscu należy zaznaczyć, że do odpowiedniego skompilowania programu w Pythonie konieczne jest zainstalowanie dwóch, niezbędnych do tego celu bibliotek. Są to kolejno:

- **requests** (<https://realpython.com/python-requests/>) - biblioteka do obsługi stron HTTP
- **beautifulsoup4** (<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>) - biblioteka do parsowania plików HTML

Konieczny jest również Python w standardzie 3.5 (podczas pracy korzystano tutaj z wersji 3.6.5).

Kod w *generate_files.py* opiera się na stronie internetowej <https://www.worldometers.info/coronavirus/country/us/>, gdzie znaleźć można aktualne statystyki dotyczące COVID-19 dla USA. Witryna jest aktualizowana co kilka godzin, a nowe dane liczbowe mogą zostać zgłoszone przez każdego użytkownika, o ile tylko podane przez niego dane można rzetelnie zweryfikować. Na tej podstawie zdecydowałem, że będzie to najlepsze źródło informacji, którego wykorzystanie nie będzie wiązało się z problematyczną obsługą API.

Now	Yesterday	Search:							
USA State	Total Cases	New Cases	Total Deaths	New Deaths	Active Cases	Tot Cases/ 1M pop	Deaths/ 1M pop	Total Tests	Tests/ 1M pop
USA Total	2,891,267	+679	132,112	+11	1,523,190	8,735	399	36,301,997	109,673
New York	420,774		32,191		289,550	21,630	1,655	4,108,129	211,176
California	252,252		6,314		176,614	6,384	160	4,448,176	112,577
Texas	190,387		2,621		94,194	6,566	90	2,273,591	78,411
Florida	178,594		3,686		148,618	8,315	172	2,083,056	96,987
New Jersey	176,455		15,270		109,188	19,866	1,719	1,485,448	167,239
Illinois	146,872		7,215		26,622	11,590	569	1,700,635	134,206
Massachusetts	109,628		8,149		8,322	15,905	1,182	946,978	137,392

Rys. 1 - Screen z tabeli Worldometer dla USA

Dla każdego z 48 stanów, które brane są pod uwagę w symulacji, z tabeli dostępnej na stronie odczytywane są (i odpowiednio formatowane z tekstu do postaci zmiennoprzecinkowej) konkretne dane liczbowe. Są to:

1. *total_cases* - liczba wszystkich dotychczasowych wystąpień koronawirusa w danym stanie USA
2. *total_deaths* - liczba wszystkich dotychczasowych śmierci z powodu koronawirusa w danym stanie USA
3. *active_cases* - liczba aktualnie chorych na COVID-19 w danym stanie USA
4. *total_tests* - liczba wszystkich testów przeprowadzonych w danym stanie USA
5. *tests_per_million* - aktualny współczynnik liczby wszystkich testów na milion mieszkańców w danym stanie USA

Ostatnie dwie z tych wartości nie będą bezpośrednio wykorzystywane. Posłużą do wyznaczenia populacji. Jest ona określona za pomocą prostego wzoru:

$$populacja = \frac{\text{liczba testów}}{\text{liczba testów na milion mieszkańców}} \cdot 1\,000\,000 \quad (1)$$

Liczbę osób wyzdrowiałych (co zaskakujące, nie jest obecna w tabeli) obliczono za pomocą wzoru:

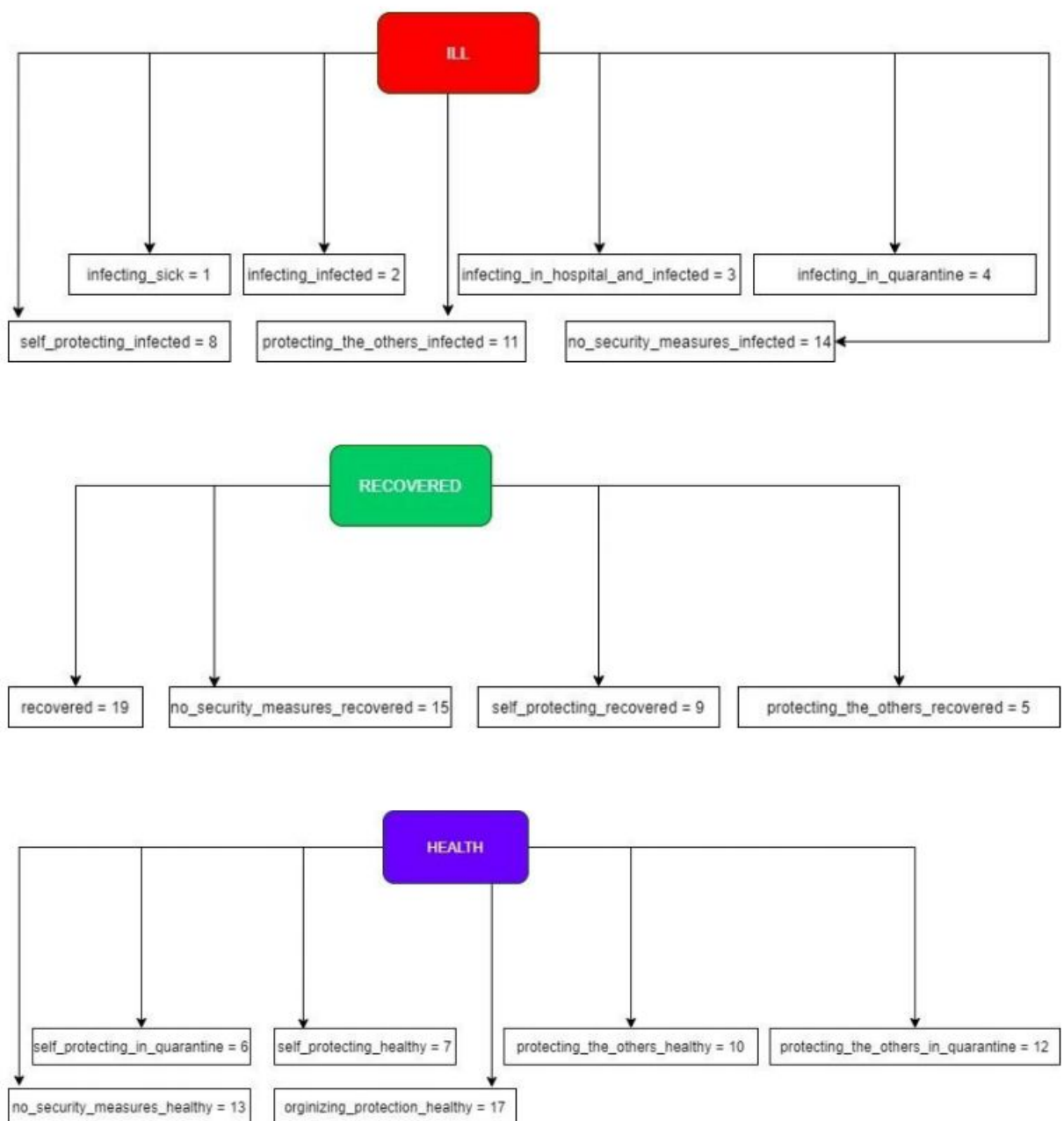
$$total\,recovered = total\,cases - total\,deaths - active\,cases \quad (2)$$

Na podstawie pobranych danych, generowane jest następnie prawdopodobieństwo wystąpienia wśród populacji danego stanu liczby osób chorych i wyzdrowiałych.

$$probability\ ill = \frac{active\ cases}{populacja} \quad (3)$$

$$probability\ recovered = \frac{total\ recovered}{populacja} \quad (4)$$

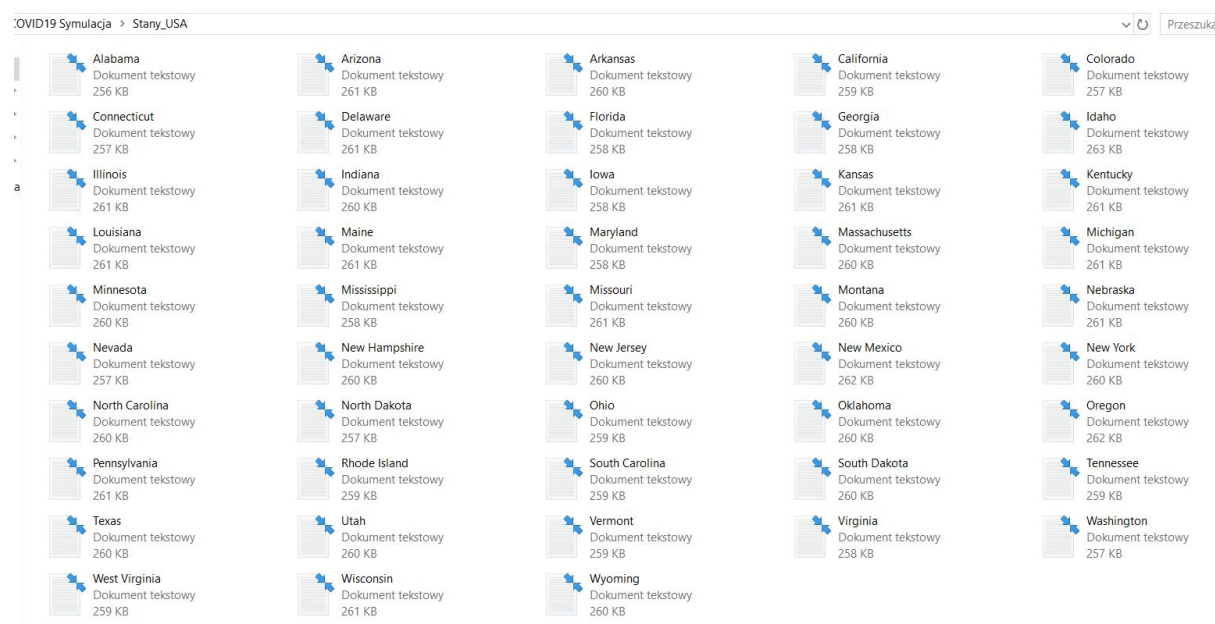
Zakłada się, że wszystkie pozostałe osoby są zdrowe, a osoby chore (jako stanowiące najmniejszy odsetek populacji) pomijamy. Do wyboru mamy aż 19 podstanów, odpowiednio przypisanych do stanów ILL, HEALTH, RECOVERED.



Stanowi DEATH przypisano domyślnie wartość 18.

Następnie program przystępuje do najważniejszego etapu: wygenerowania pliku txt zawierającego 100 000 cyfr z zakresu od 1 do 19, przedstawionych w formie listy. Prawdopodobieństwo wystąpienia w takiej liście chorych, zdrowych i wyzdrowiałych odpowiada prawdopodobieństwom wyznaczonym we wzorach (3) i (4).

Plik jest następnie zapisywany pod nazwą stanu USA, któremu od tej chwili odpowiada. Jak nietrudno się domyślić na podstawie wcześniejszych założeń, takich plików jest aż 48. Zapisywane są one w podfolderze *Stany_USA*.



Rys.2 - 48 plików txt dla poszczególnych stanów USA z listami stu tysięcy elementów

UWAGA! Jako że program korzysta ze stron internetowej, możliwe jest wystąpienie niespodziewanych błędów, jeśli administratorzy strony zdecydują się na wprowadzenie zmian w jej strukturze. W takim przypadku, tj. jeśli *generate_files.py* nie zwróci nowych plików i zwróci błędy, można zakomentować wywołującą do linijkę wewnątrz pliku *main.m* – symulacja będzie wtedy pracować na ostatnich plikach txt jakie zostały poprawnie wygenerowane. W momencie pisania tego sprawozdania, program działał prawidłowo i nie było to konieczne.

Wczytanie planszy USA 1000x1000

Podczas gdy w poprzedniej wersji mojej symulacji ograniczyłem rozmiar planszy do skromnego 200x200 (co dawało sumarycznie 40 000 komórek), tutaj zdecydowałem

się znacznie zwiększyć dokładność mapy, na której będę symulować przebieg epidemii. Uznałem, że rozmiar 1000x1000, zapewniający mi aż 1 000 000 osobnych komórek. Ponadto, rozdzielczość samej mapy była o wiele większa, co było niezbędne w celu poprawnego wyodrębnienia stanów w USA poprzez późniejszą indeksację. Początkowo rozważałem nawet rozmiar 4000x4000, jednak obliczenia na 16 000 000 komórek okazały się zbyt czasochłonne.

Plansza USA 1000x1000



Rys. 3 – mapa obszaru Stanów Zjednoczonych jako początkowa plansza o rozmiarze 1000x1000

W poprzedniej wersji mojej symulacji obszar USA był jedynie oknem, przesłaniającym obszar planszy poza konturem granic. Wprowadziłem jednak w tym aspekcie niezbędne usprawnienie, które sprawia, że komórki poza obszarem Stanów Zjednoczonych nie są uwzględniane w kolejnych obliczeniach.

Plansza zostaje poddana binaryzacji. Piksele należące do obszaru USA otrzymują początkową wartość 0 (czerni) a wszystkie pozostałe – 1 (biel). W dalszej części programu komórkom o wartości 1 zostanie przypisany nowy, dwudziesty już stan:

`non_exist = 20`. Nie będą one w żaden sposób uwzględniane w dalszych obliczeniach, co znacząco zredukuje czas wykonywania symulacji. Natomiast komórki z początkową wartością 0 zostaną na samym początku symulacji określone jako domyślnie znajdujące się w stanie `no_seacurity_measures_healthy = 13` (innymi słowy zakładamy, że przed wczytaniem danych epidemiologicznych całe USA są zdrowe).

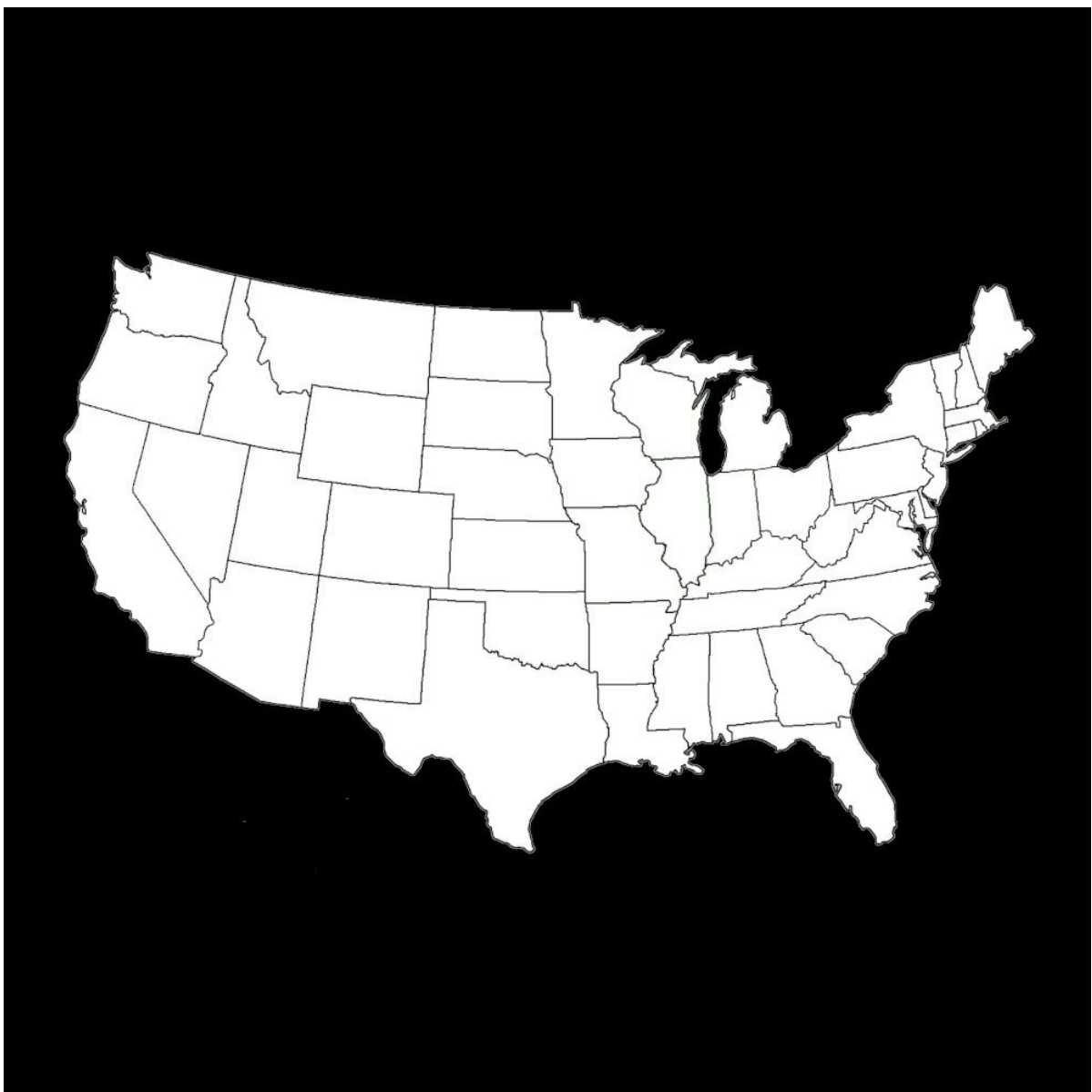
Po odrzuceniu komórek z białego tła, obszar USA na którym wykonywane będą badania składa się teraz w sumie z 272 263 komórek.

Indeksacja mapy USA – podział na stany

Kontynentalne USA podzielone są na 48 stanów, odrębnych pod względem administracyjnym, często kulturowym czy rasowym. Różnic jest bardzo wiele, więc istotne jest, by na Stany Zjednoczone nie patrzeć jak na jeden kraj, jest to bowiem federacja, w której niektóre stany mają o wiele większe znaczenie od pozostałych. W części z nich epidemia rozwinęła się bardzo mocno (jak np. w Nowym Jorku, Arizonie czy Kalifornii) a inne zaledwie tknęła (jak na przykład Montanę czy Michigan). W związku z tym, musiałem odpowiednio podzielić moją planszę na sektory, odpowiadające poszczególnym stanom. Operację przeprowadziłem wykorzystując indeksację, poznaną na Systemach Wizyjnych (przedmiot prowadzony przez dr. Piotra Pawlika i dr. Tomasza Kryjaka).

Na Rys. 4 widoczna jest identyczna co do wielkości i proporcji jak na Rys. 3 mapa USA, tyle że z naniesionymi granicami stanów. Jej rozdzielczość również wynosi 1000x1000 pikseli, co pozwoli na odpowiednie wyodrębnienie wszystkich poszukiwanych regionów, bez żadnych strat. Gdyby rozdzielczość była mniejsza, takie regiony jak Rhode Island czy Connecticut mogłyby zostać niewykryte przez algorytm indeksacji Matlaba. Oczywiście, przed wykonaniem indeksacji konieczna była binaryzacja mapy, poprzez funkcję `im2bw` (próg ustalono na 240/255). Indeksację przeprowadzono za pomocą operacji wbudowanej w Matlaba `bwlabel` (za argument przyjęto 8, czyli otoczenie Moore'a).

Indeksacja została przeprowadzona w pliku `wydzielanie_stanow.m`, gdzie konieczne okazało się odrzucenie kilkunastu obiektów (na skutek występowania pojedynczych czarnych pikseli na obrzeżach USA, indeksacja wykryła początkowo aż 65 obiektów!). Usunięto więc te, które składały się z mniej niż 40 pikseli, co pozwoliło na uzyskanie samych poprawnych i rzeczywistych obiektów.



Rys. 4 - podział USA na stany, plik wykorzystany do przeprowadzenia indeksacji.

Na skutek indeksacji wyodrębniono 51 obszarów. Jak widać, jest to aż o 3 za dużo, jednak przyczyna takiego stanu rzeczy zostanie wyjaśniona poniżej.

Indeks	Obszar
1	California
2	Oregon
3	Washington
4	Nevada

5	Arizona
6	Idaho
7	Utah
8	Montana
9	Wyoming
10	New Mexico
11	Colorado
12	Texas
13	Nebraska
14	South Dakota
15	North Dakota
16	Oklahoma
17	Kansas
18	Minnesota
19	Iowa
20	Missouri
21	Arkansas
22	Louisiana
23	Wisconsin
24	Illinois
25	Mississippi
26	Półwysep Górny Michigan
27	Tennessee
28	Kentucky
29	Indiana
30	Alabama
31	Półwysep Dolny Michigan

32	Florida
33	Ohio
34	Georgia
35	North Carolina
36	Virginia
37	South Dakota
38	West Virginia
39	Pennsylvania
40	New York
41	Zachód Maryland
42	Wschód Maryland
43	Delaware
44	New Jersey
45	Fragment Delaware
46	Vermont
47	Massachusetts
48	Connecticut
49	New Hampshire
50	Maine
51	Rhode Island

Zaznaczone tu na żółto obszary są oddzielone od głównego obszaru stanu, do którego przynależą. Najlepiej widać to na przykładzie Michigan, podzielonego na dwa półwyspy:



Rys. 5 - stan Michigan podzielony na Półwysep Dolny i Górny

W związku z takim stanem rzeczy, obszary zaznaczone na żółto trzeba było następnie przypisać ręcznie do ich macierzystych terenów na mapie USA.

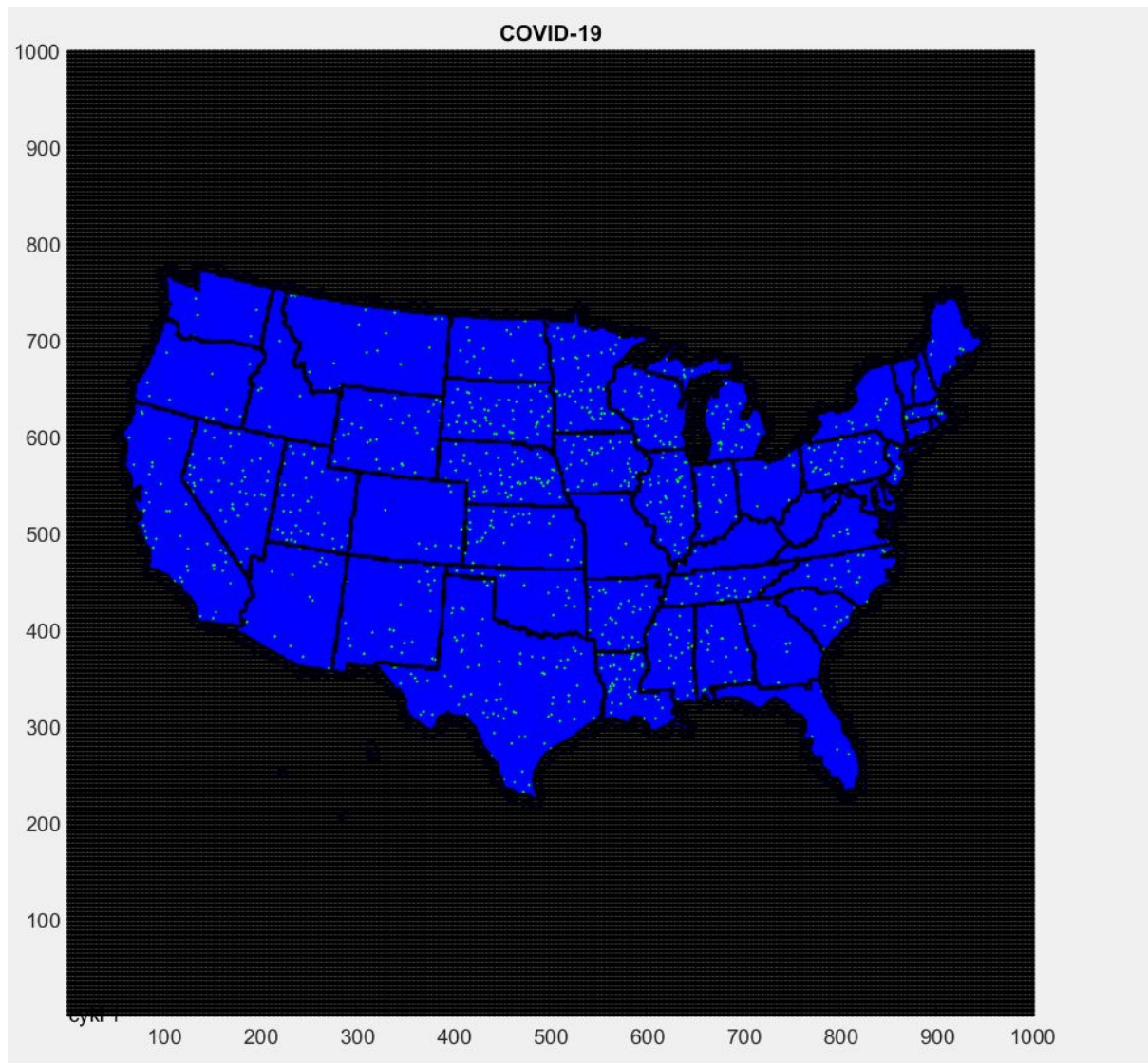
Nanoszenie początku epidemii na obszary stanów USA

Funkcja `generuj_poczatek.m` odpowiada za poprawne przeniesienie zawartości utworzonych w Pythonie plików txt na obszary poszczególnych stanów USA. Dla kolejnych pikseli poszczególnych stanów odczytywane są kolejne elementy z liczącej 100 000 elementów listy zawartej w pliku tekstowym. Jako że są tam ułożone w kompletnie losowej kolejności, każdy stan USA staje się w tym momencie odrębnym regionem, z innym zagęszczeniem osób chorych i wyzdrowiałych.

`generate_files.m` ma bardzo złożoną strukturę i do odpowiedniej obsługi plików wykorzystuje takie mechanizmy jak matlabowe komórki (cell) oraz obsługę wielu obszernych list.

Wyjaśnić w tym momencie można, dlaczego listy tworzone w `generate_files.py` mają aż 100 000 elementów. Tak duże liczby są konieczne, ponieważ stany na mapie USA składają się z kilkudziesięciu tysięcy komórek (dla przykładu Teksas liczy ich aż 24 665). Jako że na każdą taką komórkę przypada jeden element z listy zawartej w pliku txt, musi mieć ona odpowiednią długość. Oczywiście, mogłaby być nieco krótsza (zakres 25 000 elementów w zupełności by wystarczył) jednak taka

zmiana nie wpływa znacząco na czas wykonywania programu i jest on niewiele mniejszy.



Rys. 6 - efekt naniesienia na mapę USA wartości z plików txt, czyli wizualizacja pierwszego cyklu

Na Rys. 6 widać wyraźnie różnice w zagęszczeniu przypadków w poszczególnych stanach. Najbardziej widoczny jest kolor zielony (wyzdrowiali) pomimo, że liczba chorych (czerwony) jest zazwyczaj niewiele mniejsza bądź równa. Większe kontrasty pomiędzy poszczególnymi stanami będą się uwidaczniać w kolejnych cyklach.

Granice stanów, zostały naniesione na mapę jako czarne linie, zakrywające znajdujące się od nimi komórki. Owe komórki nie przynależą do żadnego ze stanów USA, ale mogą przyjmować wszystkie stany zdrowotne (od 1 do 19), dzięki czemu choroba może się roznosić na nowe obszary.

Zmiana stanów komórek na obszarze stanów USA

W tym momencie przechodzimy do najważniejszej części całego programu, czyli właściwej symulacji. Po naniesieniu na mapę stanów USA odpowiednich wartości wygenerowanych w Pythonie oraz przypisaniu wszystkim komórkom poza obszarem Stanów Zjednoczonych wartości 20 (non_exist) przystępujemy do zmieniania stanów poszczególnych komórek.

Jak już wspomniane było wcześniej, każda komórka może przyjąć jeden z 19 możliwych podstanów (dokładne ich opisanie w paragrafie „Generowanie plików wejściowych - Python”). Określić trzeba jeszcze jak wyglądają możliwe przejścia pomiędzy poszczególnymi podstanami. Zostało to zaprezentowane na poniższej tabeli:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
2	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
4	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
5	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
6	0	1	0	1	0	1	1	1	0	1	1	1	0	0	0	1	1	0	0	0
7	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0
8	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
11	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
12	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
13	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0
14	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0	0	1	0
15	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
16	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
17	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
19	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	0	1	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Rys. 7 - tabela sąsiedztwa dla wszystkich podstanów. 1 oznacza możliwość przejścia.

Po pierwszym cyklu, dla każdej z badanych komórek następuje sprawdzenie stanu sąsiedztwa, za pomocą funkcji *fPoliczSasiadow.m* oraz *stany_sasiedzi.m*. Sprawdzana tu jest liczba wystąpień każdego z możliwych podstanów wokół badanej komórki. Operacja jest oparta o sąsiedztwo ośmioelementowe, czyli sąsiedztwo Moore'a. Po długich przemyśleniach zdecydowałem się nie korzystać z proponowanego mi sąsiedztwa von Neumanna, uznając je za mniej precyzyjne, zwłaszcza w przypadku tak dużej liczby obiektów.

Dla każdej komórki tworzona jest dwudziestoelementowa lista *sasiedzi*, do której zapisywana jest liczba wystąpień każdego z możliwych podstawów. Następnie na jej podstawie wyliczana jest zmienna *stan_sasiedztwa*:

$$stan_sasiedztwa = \sum_{i=1}^{20} a_i \cdot sasiedzi(i) \quad (5)$$

Gdzie kolejno:

sasiedzi = [*infecting_sick*, *infecting_infected*,
infecting_in_hospital_and_infected, *infecting_in_quarantine*,
protecting_the_others_recovered, *self_protecting_in_quarantine*,
self_protecting_healthy, *self_protecting_infected*,
self_protecting_recovered, *protecting_the_others_healthy*,
protecting_the_others_infected,
protecting_the_others_in_quarantine,
no_security_measures_healthy, *no_security_measures_infected*,
no_security_measures_recovered, *organizing_protection_infected*,
organizing_protection_healthy, *dead*, *recovered*, *non_exist*]

Współczynniki *a* obecne w równaniu (5) zostały dobrane na drodze eksperymentalnej, bo wielu próbach i eksperymentach, tak, by wyniki symulacji były jak najbardziej zbliżone do stanu rzeczywistego i jednocześnie widoczne (tzn. by na przykład całe społeczeństwo nie wyzdrowiało, zachorowało lub zmarło w ciągu kilku cykli).

<i>i</i>	1	2	3	4	5	6	7	8	9	10
<i>a_i</i>	-10.5	-10.5	-10.5	-10.5	1.2	0.8	0.8	-1.5	1.2	1.2

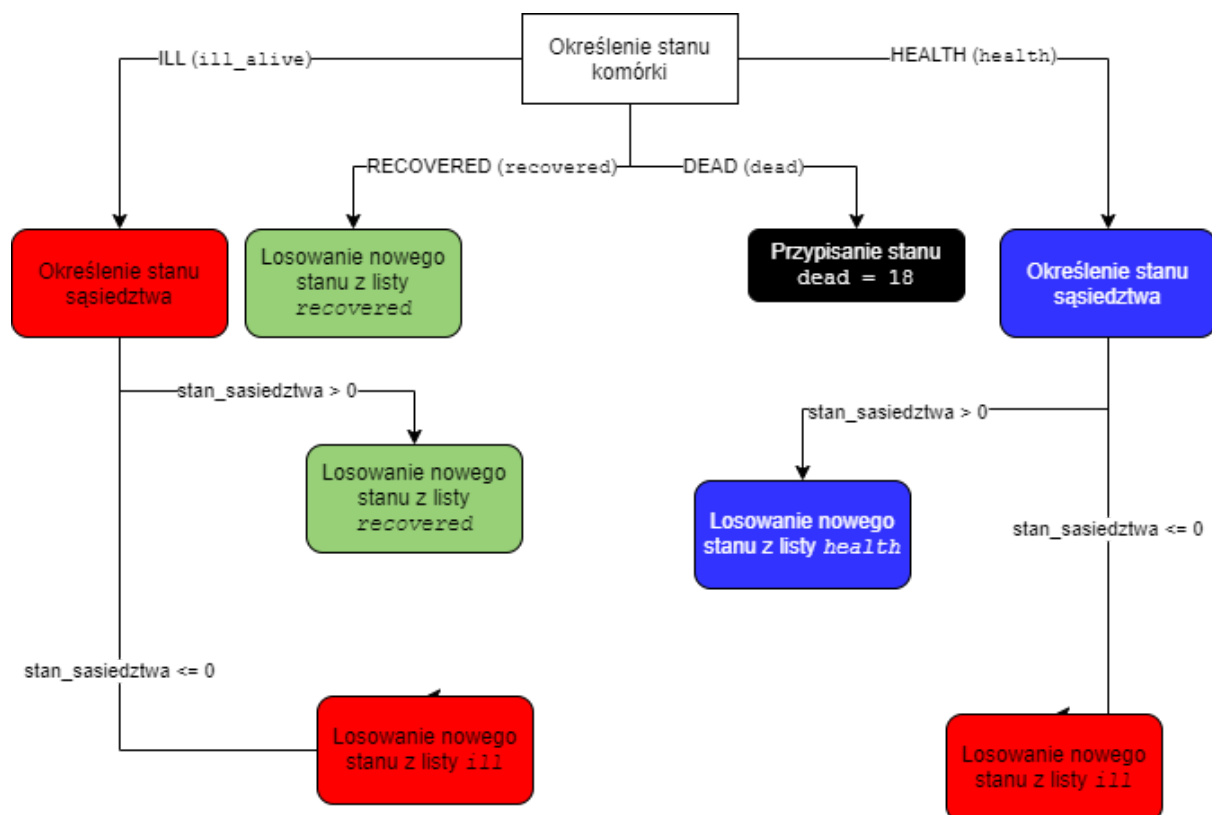
<i>i</i>	11	12	13	14	15	16	17	18	19	20
<i>a_i</i>	-5	1.2	0.5	-10.5	2	-10.5	2	0	2	0

Jak widać, założono, że osoba zmarła *dead* = 18 nie ma żadnego wpływu na swoje bezpośrednie otoczenie.

Następnie korzystamy z funkcji *zmiana_komorki.m*. To tutaj podejmowana jest ostateczna decyzja o losie komórki w kolejnym cyklu. Wszystko zależy od wartości

stan_sasiedztwa, której wyliczanie opisane zostało powyżej. Tworzymy teraz cztery listy, z których będziemy korzystać w celu określenia czy dany obiekt powinien pozostać zdrowy, zachorować, chorować dalej, wyzdrowieć bądź umrzeć.

```
ill = [1 2 3 4 8 11 14 16 18];
ill_alive = [1 2 3 4 8 11 14 16];
health = [6 7 10 12 13 17];
recovered = [5 9 15 19];
dead = [18];
```

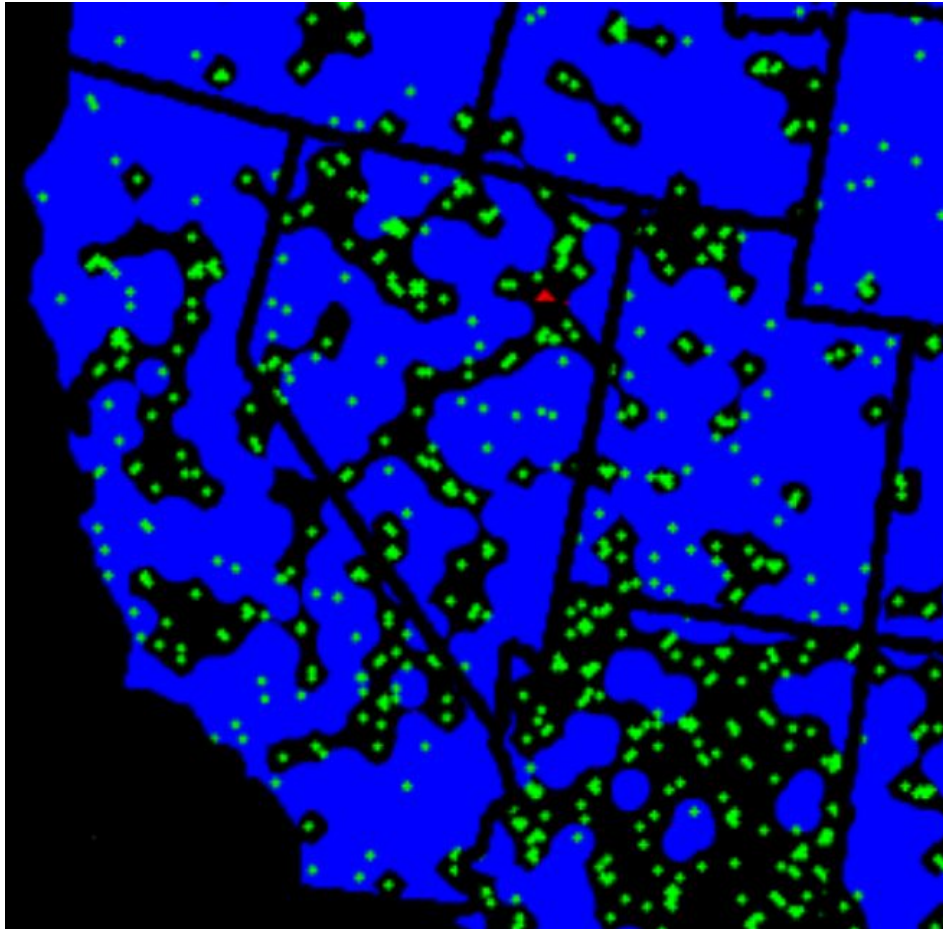


Jak więc widać, możliwe jest przejście z jednego stanu do drugiego w ciągu zaledwie jednego cyklu. W związku z tym jeden cykl odpowiada okresowi mniej więcej dwóch tygodni, jako że tyle trwa mniej więcej okres inkubacji wirusa, wyzdrowienia, zachorowania bądź utraty życia.

Widoczne jest tu również przyjęte przeze mnie założenie, wedle którego osoba która już wyzdrowiała nie może ponownie zachorować i już na zawsze zachowa odporność na badaną chorobę.

W ten oto sposób określana jest nowa macierz epidemiologiczna, nanoszona następnie na mapę Stanów Zjednoczonych. Należy przy tym pamiętać, że w tym

momencie osoby chore mogą już zarażać komórki z innych stanów, dzięki faktowi, że komórki ukryte pod granicami stanów również mogą się zarazić (bądź wyzdrowieć).



Rys. 8 widoczne rozprzestrzenianie się epidemii na zachodnim wybrzeżu USA, również poza granice poszczególnych stanów. Widoczne jest również największe ognisko (Arizona).

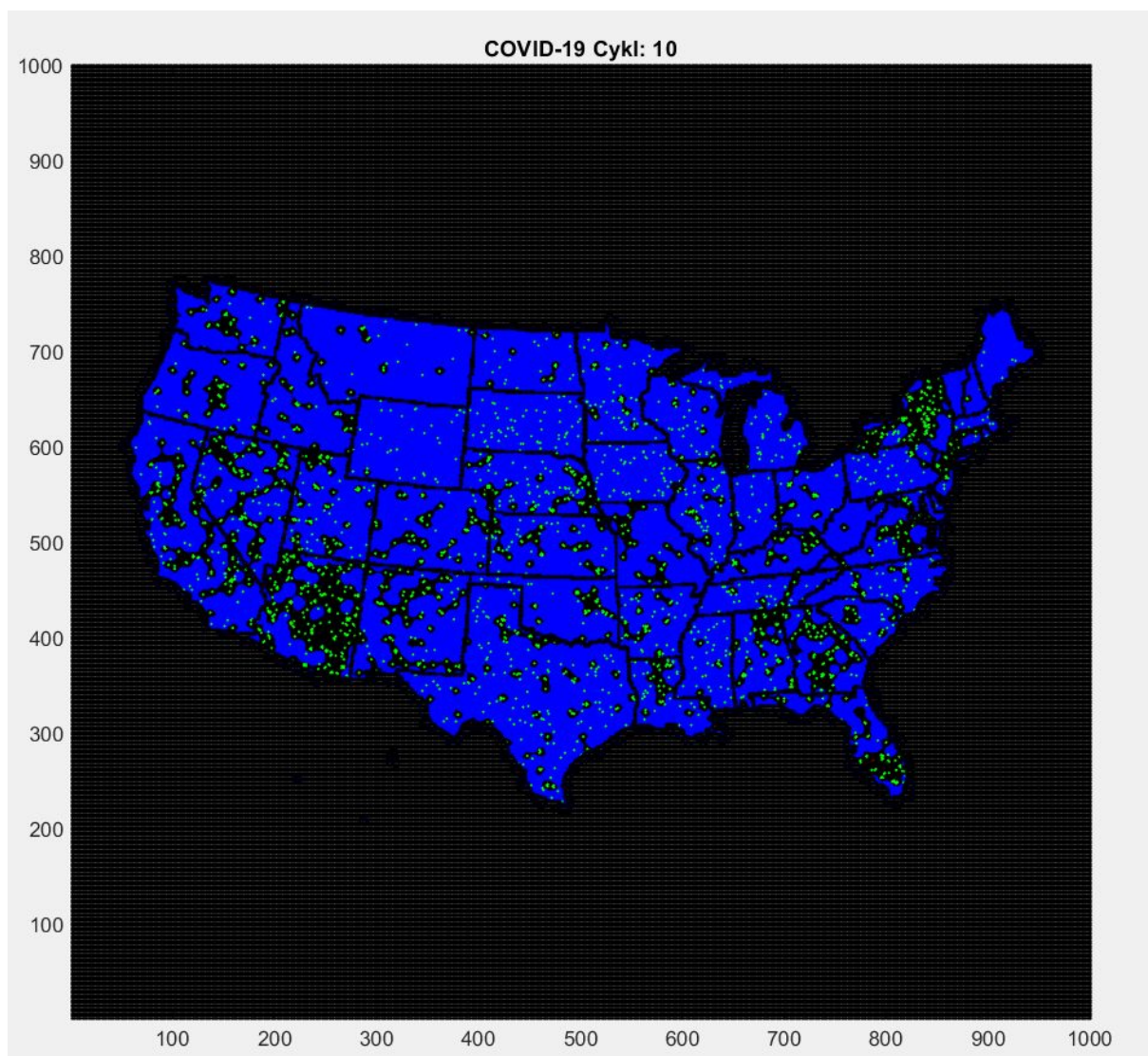
Aktualna sytuacja każdego ze stanów USA jest zapisywana w zmiennej `tablica_stanow_usa`, która przechowuje informacje o liczbie chorych, zdrowych, wyzdrowiałych i martwych dla każdego ze stanów w określony poniżej sposób w każdej z 51 komórek cell:

`[chorzy, zdrowi, wyzdrowiali, martwi]`

Nastąpiło to nie tylko dzięki zmianie rozmiarów planszy ale również znacznej modyfikacji równania określającego stan sąsiedztwa oraz podział obszaru USA na poszczególne, odrębne stany.

Podczas symulacji wyraźnie widać, że kilka regionów narażonych jest na o wiele większe ryzyko wystąpienia śmierci bądź zachorowania z powodu wystąpienia koronawirusa. Są to głównie obszary przynależne do stanów Nowy Jork, Arizona, California i tereny Pasa Biblijnego.

Wyraźnie widać, jak epidemia wkracza na kolejne obszary, zarażając i uśmiercając kolejne osoby. Granice stanów nie stanowią żadnej przeszkody, niejednokrotnie można wręcz dostrzec struktury, w jakie układają się całe obszary osó zarażonych i martwych (głównie na zachodnim wybrzeżu).



Rys. 11 wyraźny rozwój epidemii przez 10 cykli. Widać obszary najbardziej skażone (Arizona, Nowy Jork) oraz takie, które nadal pozostają stosunkowo bezpieczne (Delaware, Michigan, Wyoming).

Powiększają się też jednak skupiska osób wyzdrowiałych, zwiększających potem szansę na powrót do zdrowia dla nowych skażonych wirusem komórek. W momencie gdy liczba zmarłych i chorych zacznie jednak gwałtownie rosnać, tempo przyrostu osób wyzdrowiałych zacznie gwałtownie maleć, co widoczne będzie na mapie oraz w danych liczbowych, podawanych w konsoli Matlaba po każdym zakończonym cyklu.

Przykładowy rozwój epidemii, dla danych pobranych z witryny internetowej 4 lipca 2020 o godzinie 23:05 zamieściłem na końcu niniejszego sprawozdania w formie dziesięciu plansz, dla każdego kolejnego cyklu (próba wyniosła tutaj 10 cykli). Są one również generowane przez sam program i zapisywane w formacie *pdf* w folderze projektu. Niestety, format *jpg* nie pozwala dokładnie przyjrzeć się mapie, podobnie jak zamieszczona kilka linii wyżej jej miniaturka. Ponieważ naniesienie legendy na wykres powodowało błędy, pozwolę sobie przypomnieć jej formę:

- niebieski - zdrowi
- zielony - wyzdrowiali
- czerwony - chorzy
- czarny - zmarli

3. Interfejs użytkownika - GUI

Do projektowania GUI przystąpiłem wraz z grupą w składzie: Marta Sosenkiewicz, Aleksander Siwek, Karol Stryczek. Udało nam się stworzyć funkcjonalny interfejs użytkownika, korzystający z działania naszych symulacji dla Norwegii, Australii, Francji oraz, w moim przypadku - Stanów Zjednoczonych.

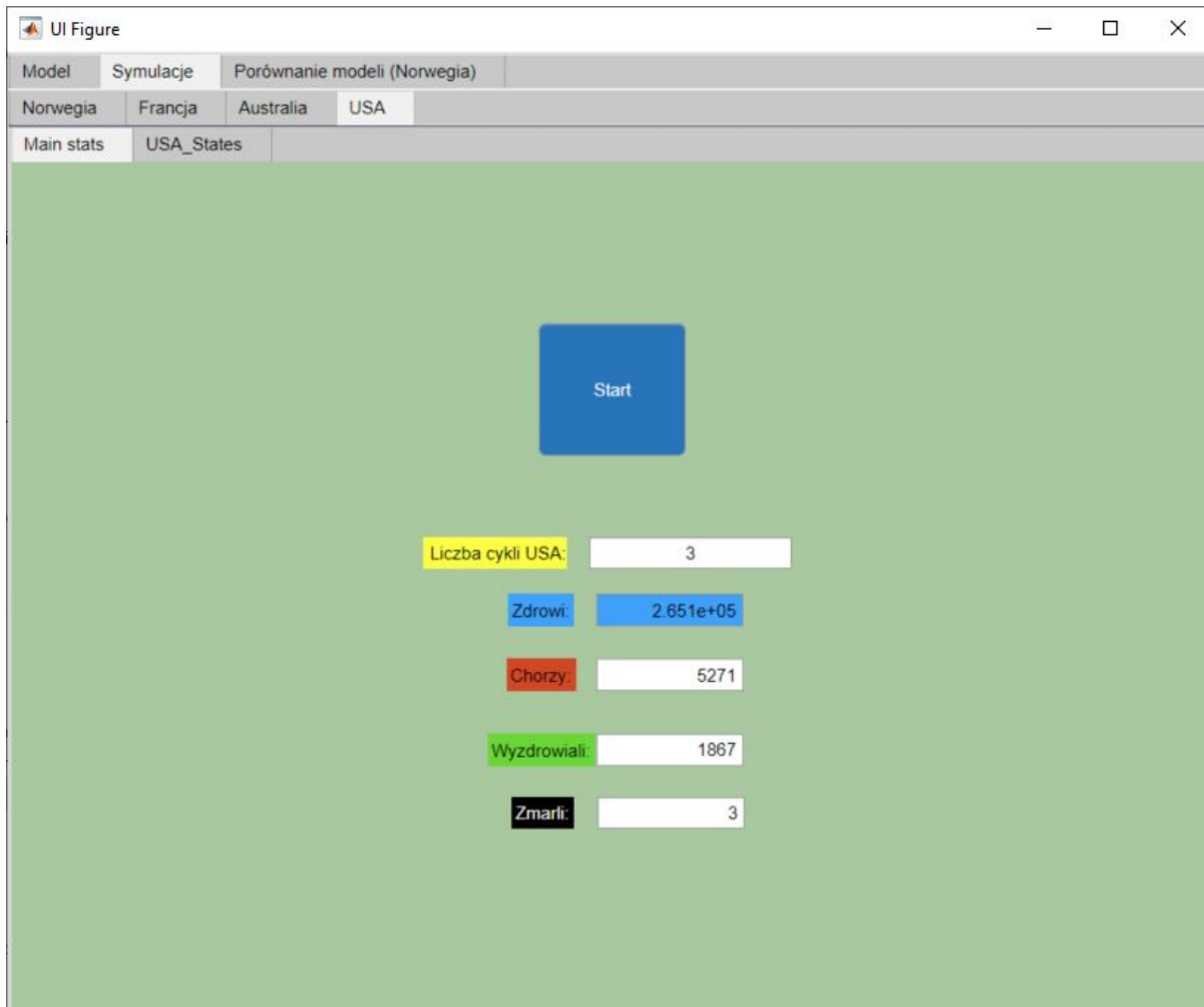
W wersji przystosowanej do GUI występują jednak pewne zasadnicze różnice w stosunku do oryginalnej formy symulacji, która nie współpracuje z App Designer Matlaba.

Wersja oryginalna symulacji	GUI
Pliki txt zapisane w podfolderze Stany_USA	Pliki txt zapisane w głównym folderze projektu
<i>main.m</i> jako plik uruchamiający symulację	<i>main(app)</i> jako funkcja wywoływana w GUI
komenda <i>clear all</i> na początku programu	komenda <i>clear all</i> usunięta (powodowała usunięcie argumentów podanych w GUI)

plik `generate_files.py` umieszczony w podfolderze `Python_workspace` i bezproblemowo wywoływany przez Matlaba funkcją `system()`

plik `generate_files.py` umieszczony w głównym folderze projektu **z koniecznością podania bezpośredniej ścieżki na komputerze do pliku (4 linijka funkcji `main(app)`)**

W przypadku problemów z działaniem pliku w Pythonie, zaleca się po prostu zakomentować wywołującą go linijkę w `main(app)` i przeprowadzić symulację na dostarczonych już plikach `txt`.



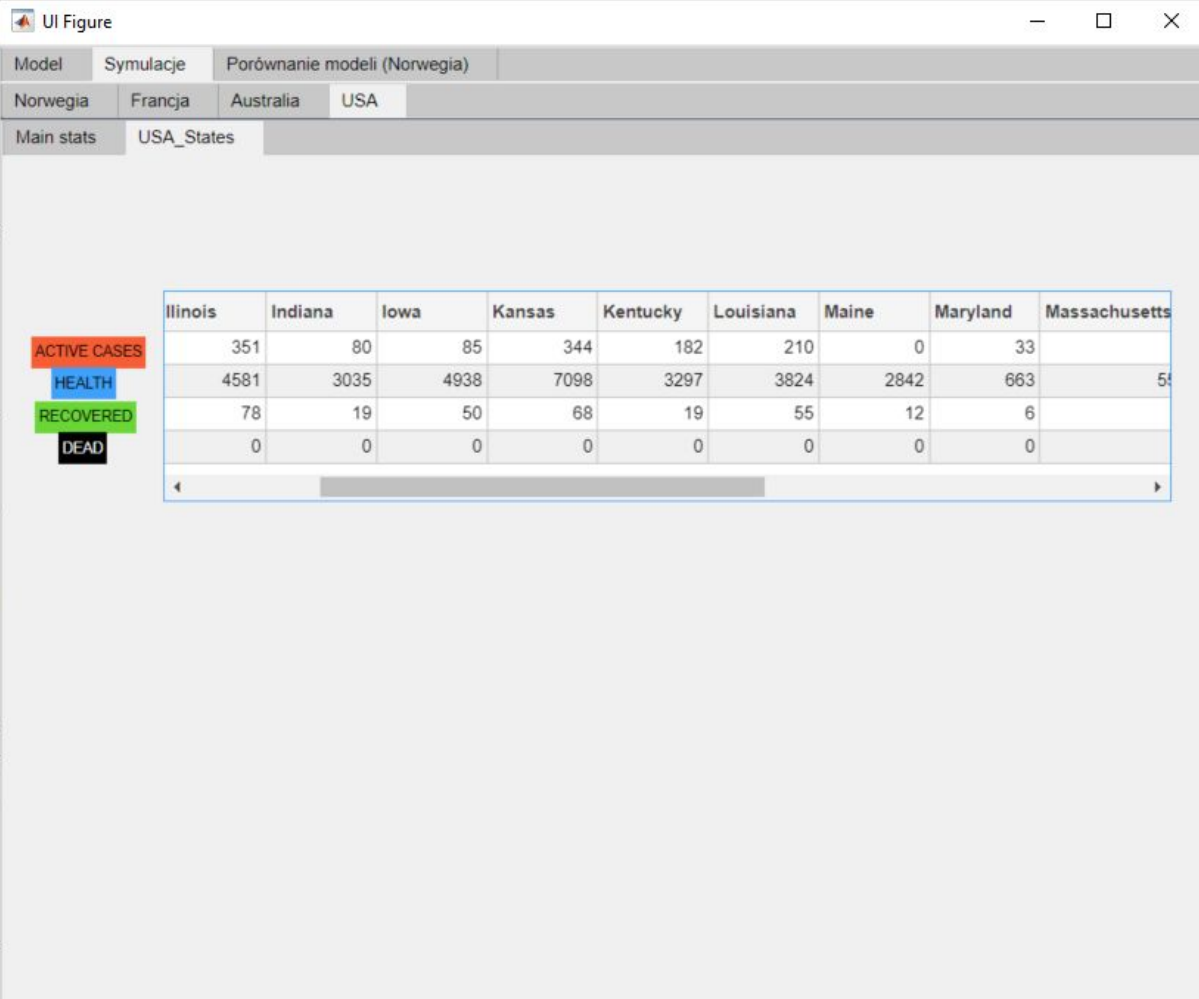
Rys. 12 główny interfejs aplikacji dla symulacji USA

W głównym interfejsie aplikacyjnym dla USA podajemy liczbę cykli (*Liczba cykli USA*), dla których wykonana ma być symulacja. W celu zaobserwowania ciekawych i miarodajnych wyników, zaleca się wykonanie przynajmniej czterech cykli.

UWAGA! Jeden cykl może się wykonywać nawet kilka minut!

Po wciśnięciu przycisku `Start` wywoływana jest funkcja `main(app)`, która rozpoczyna wykonywanie symulacji.

W oknach *Zdrowi*, *Chorzy*, *Wyzdrowiali* i *Zmarli* wyświetlane są aktualne dane dla całego obszaru USA, czyli sumaryczne liczby ze wszystkich czterdziestu ośmiu stanów.



	Illinois	Indiana	Iowa	Kansas	Kentucky	Louisiana	Maine	Maryland	Massachusetts
ACTIVE CASES	351	80	85	344	182	210	0	33	
HEALTH	4581	3035	4938	7098	3297	3824	2842	663	5
RECOVERED	78	19	50	68	19	55	12	6	
DEAD	0	0	0	0	0	0	0	0	

Rys. 13 - *USA_States* czyli alfabetyczna lista statystyk dla poszczególnych stanów USA

W okienku *USA_States* można natomiast podejrzeć aktualną sytuację w każdym stanie. W celu wygenerowania takiej tabeli konieczne było również stworzenie całkowicie nowej funkcji *tworzenie_wiersza(tablica_stanow_usa, stan)* która zwraca cztery listy komórkowe: chorych, zdrowych, wyzdrowiałych oraz zmarłych dla każdego ze stanów USA.

4. Podsumowanie

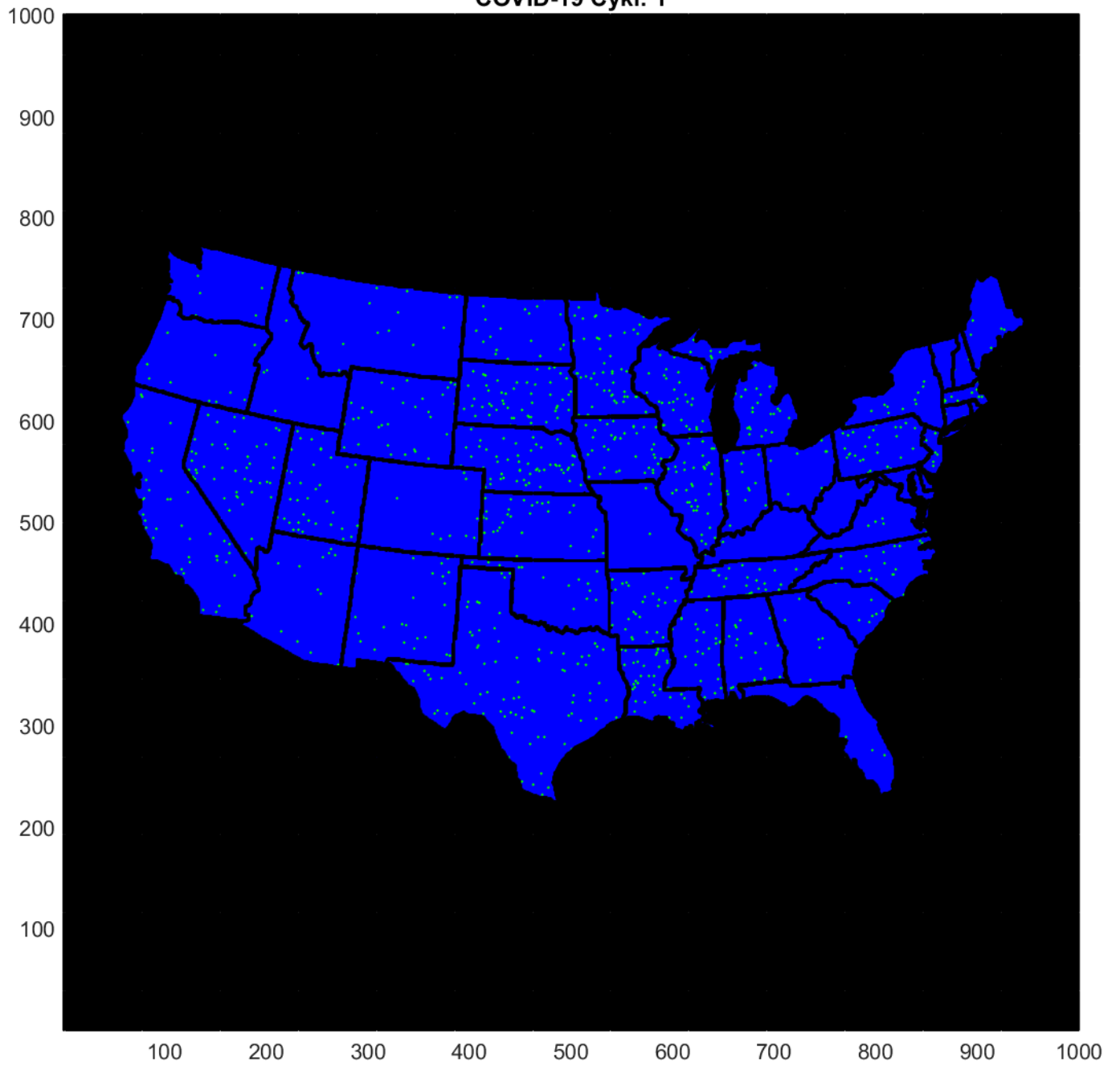
Podczas wykonywania tego ćwiczenia mogłem pozwolić sobie na nieco większą kreatywność w podejściu do wielu problemów niż inni studenci, realizujący to samo

zadanie dla innych krajów. Dzięki temu mogłem skupić się na dzieleniu USA na poszczególne regiony i opracowywanie dla nich odpowiedniego działania symulacji. Udało mi się pozbyć niepotrzebnych obliczeń dla tła, występujących w poprzedniej wersji programu, wyeliminować prostokątne figury na mapie oraz wykorzystać dane pobrane z witryny internetowej w celu wygenerowania odpowiedniego stanu początkowego dla epidemii.

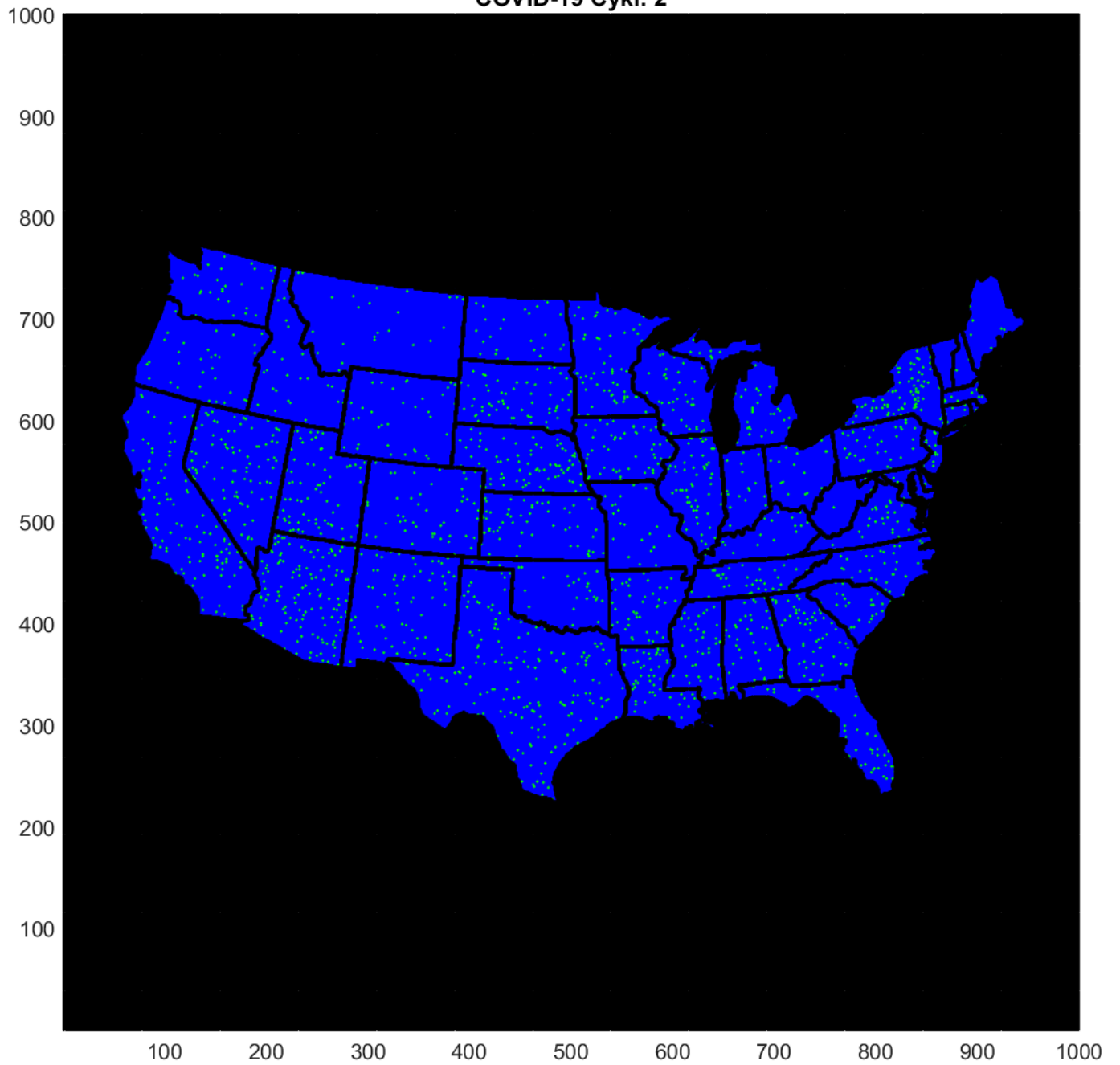
Ćwiczenie pozwoliło mi na dokładniejsze poznanie środowiska Matlaba, oraz wykorzystanie jego możliwości w większym stopniu niż na innych laboratoriach. Udało mi się dzięki temu stworzyć miarodajną symulację rzeczywistego rozwoju epidemiologicznego na terenie Stanów Zjednoczonych Ameryki, które można łatwo przedstawić każdemu użytkownikowi ze względu na czytelne wyniki w formie mapy z odpowiednimi granicami. W programie występuje co prawda element losowości, jednak jest on i tak oparty na danych ściągniętych z zaufanej witryny statystycznej.

Symulacja nie jest pozbawiona wad i jest w niej wiele rzeczy, które mogłyby zostać w przyszłości poprawione, jak na przykład czas, jaki przypada na jeden cykl programu (kilkanaście dni może się okazać za mało dokładne) jednak na potrzeby realizacji tego ćwiczenia udało mi się dostarczyć funkcjonalną symulację, mogącą w ograniczonym stopniu przewidzieć rozwój epidemii na podzielonym na konkretne tereny obszarze.

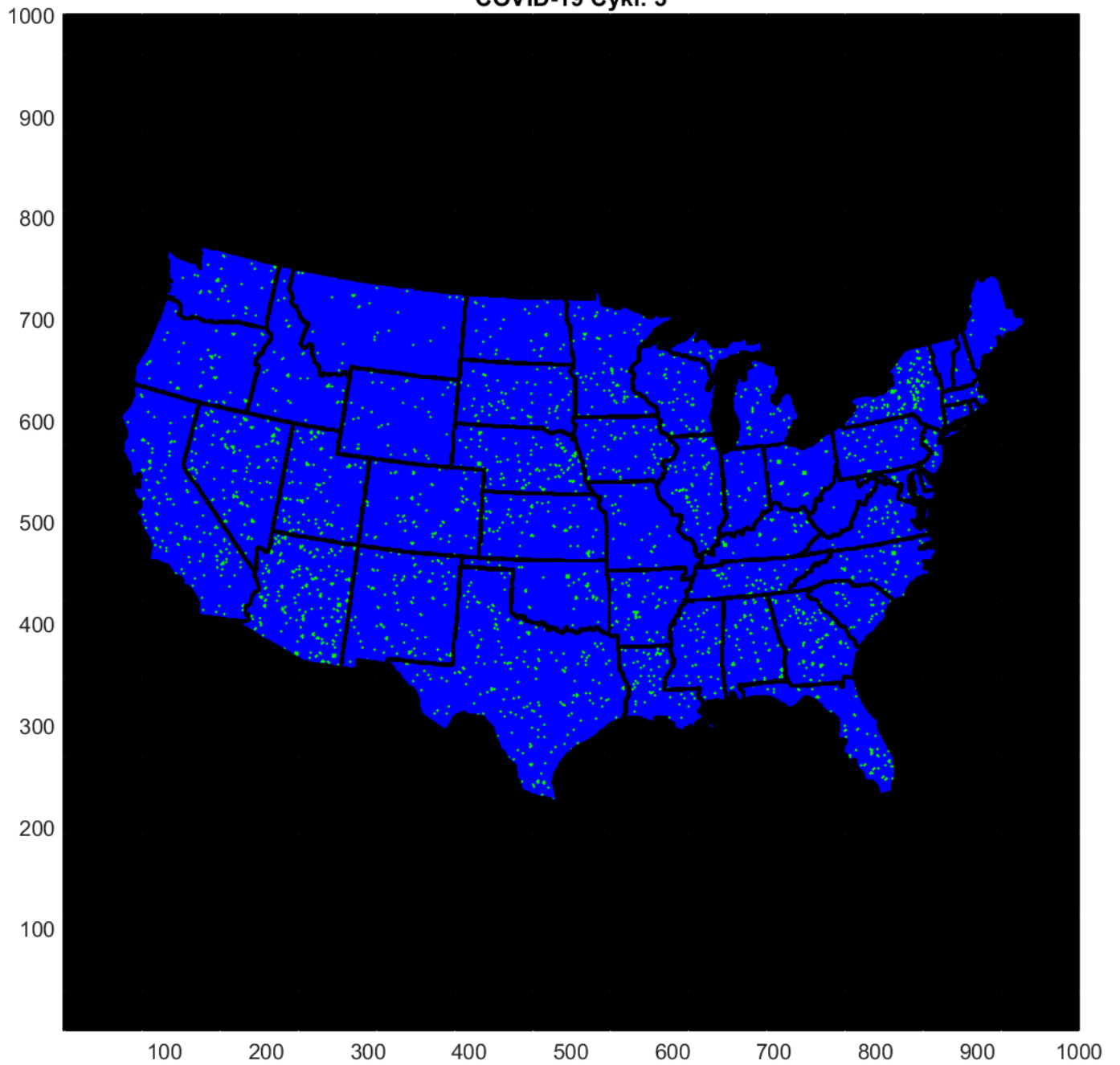
COVID-19 Cykl: 1



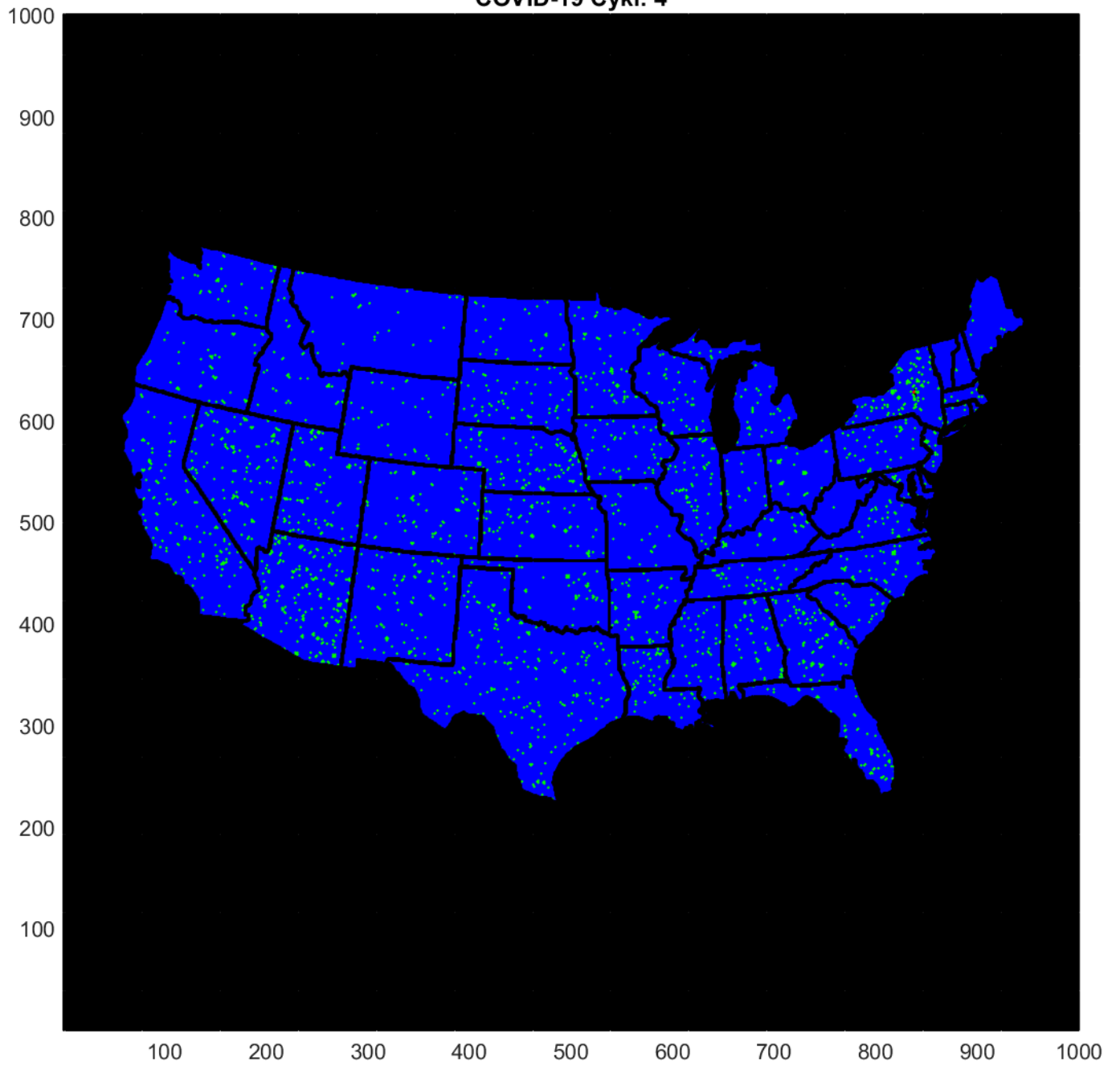
COVID-19 Cykl: 2



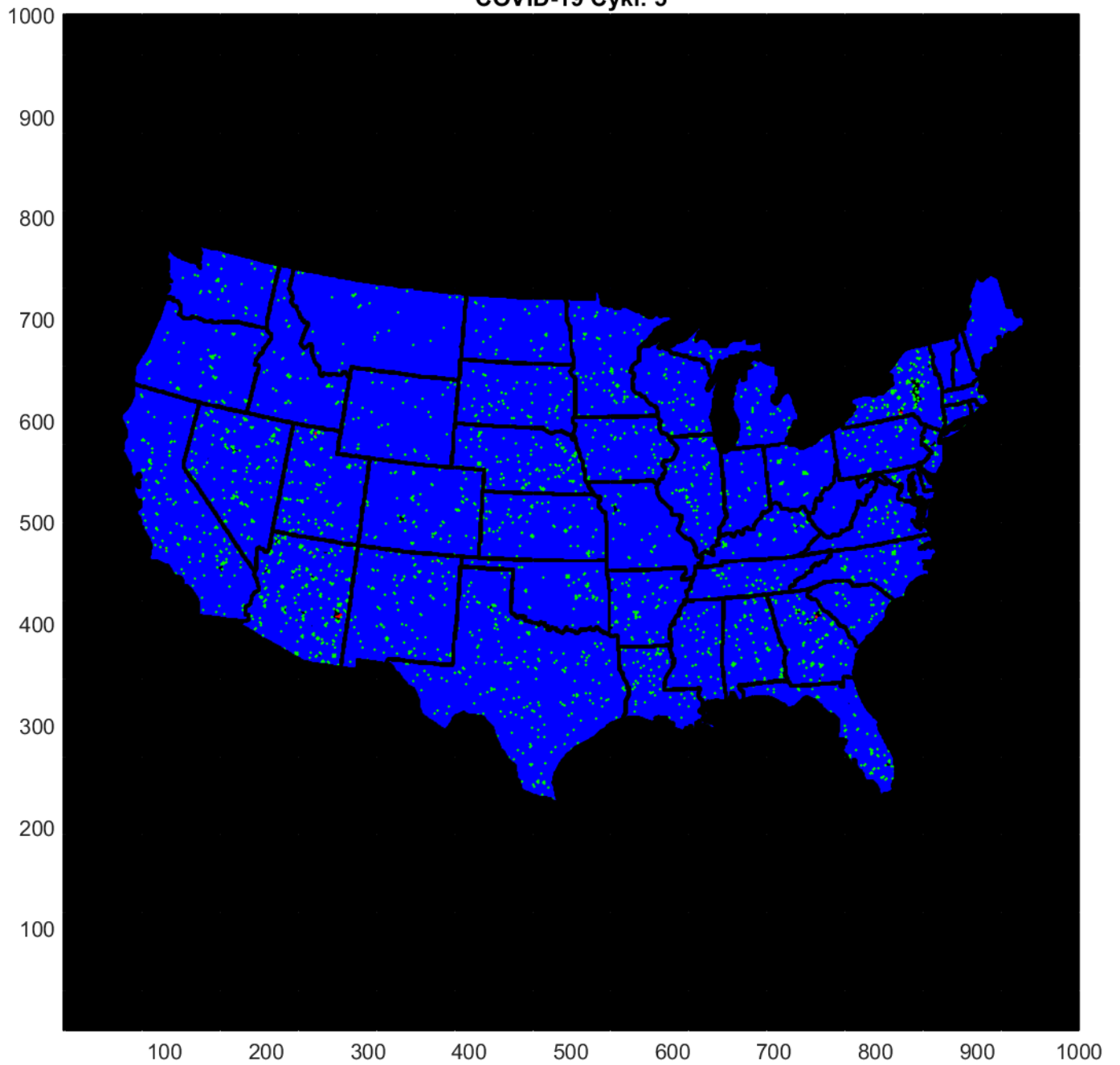
COVID-19 Cykl: 3



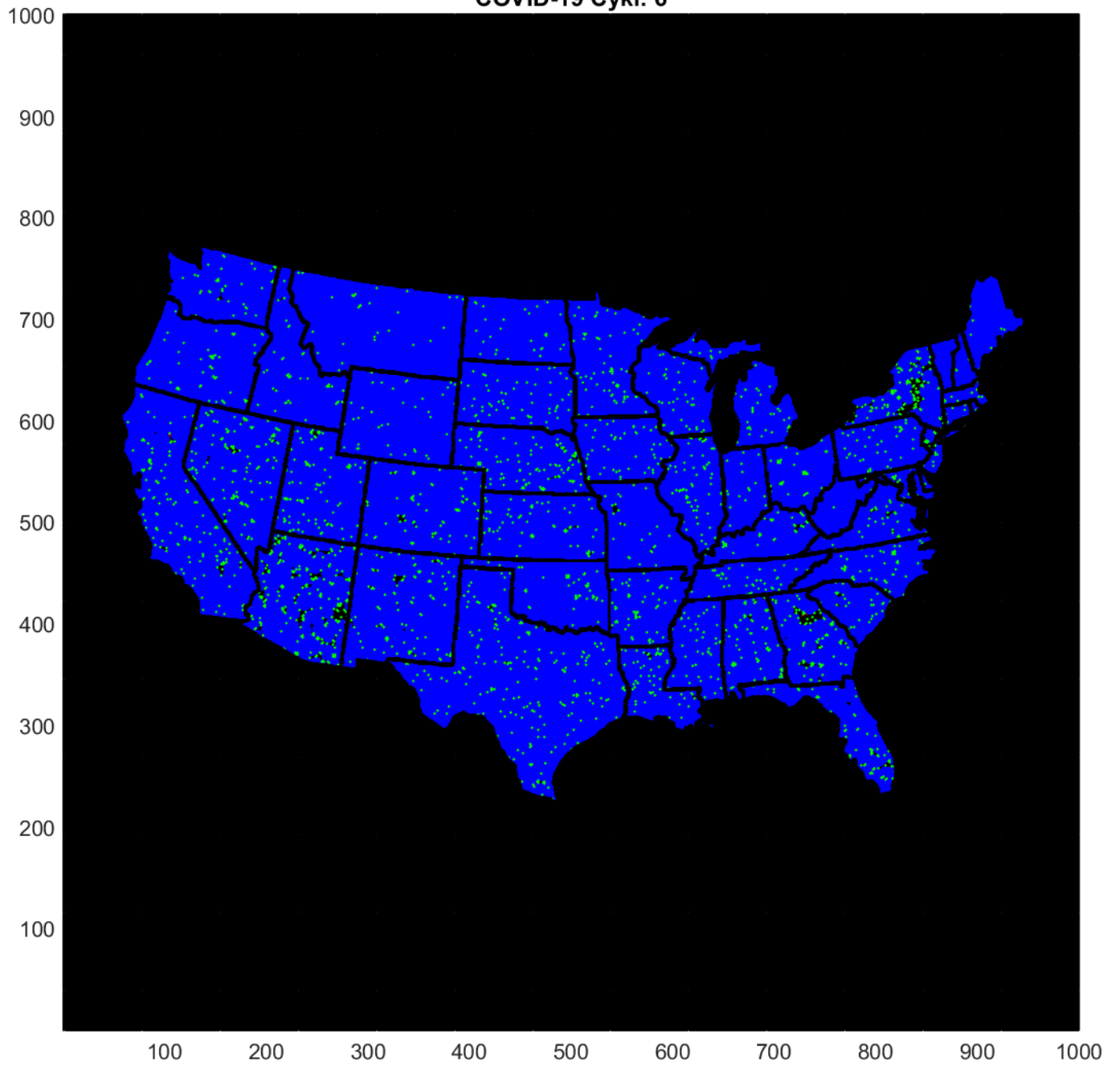
COVID-19 Cykl: 4



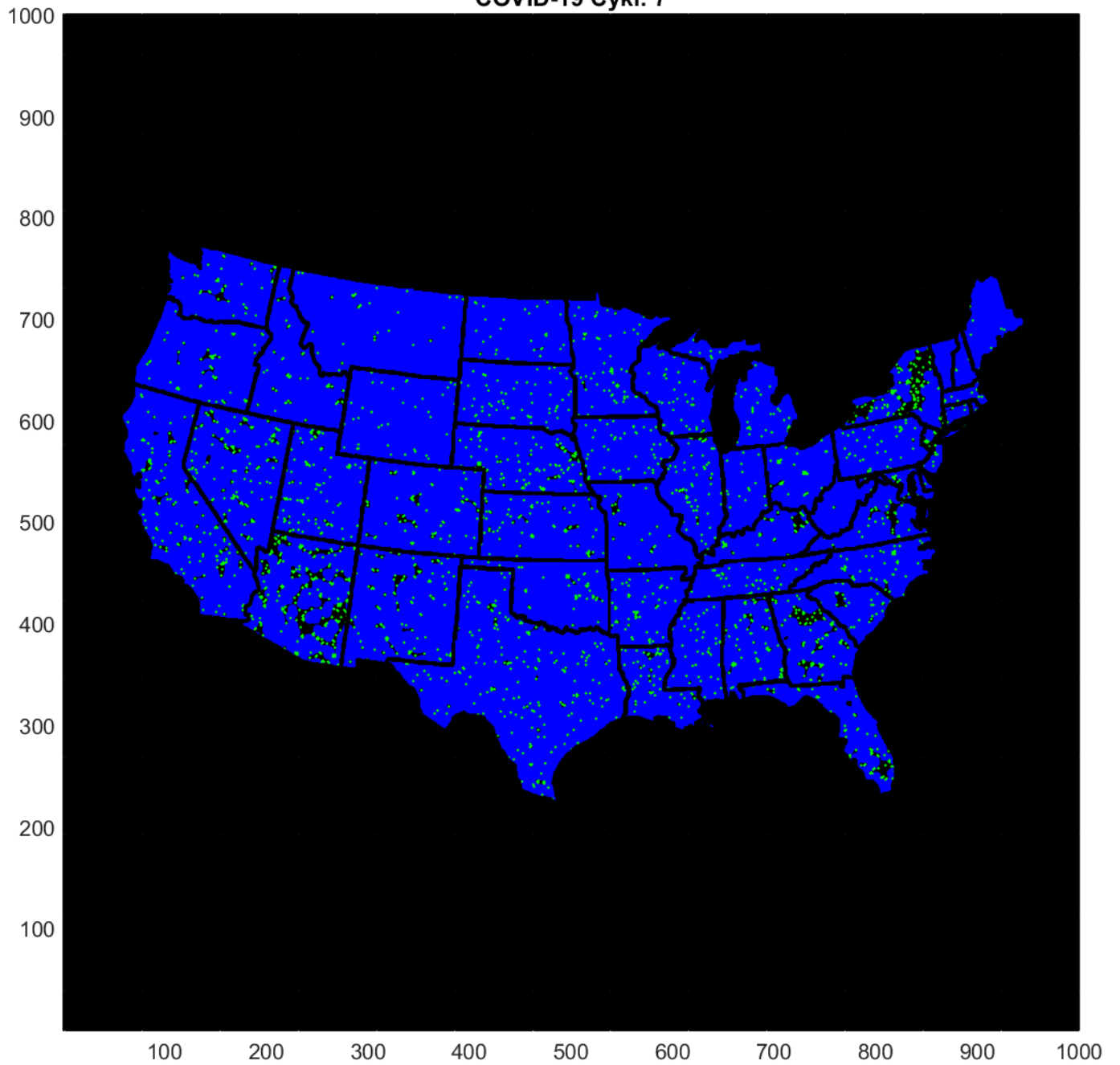
COVID-19 Cykl: 5



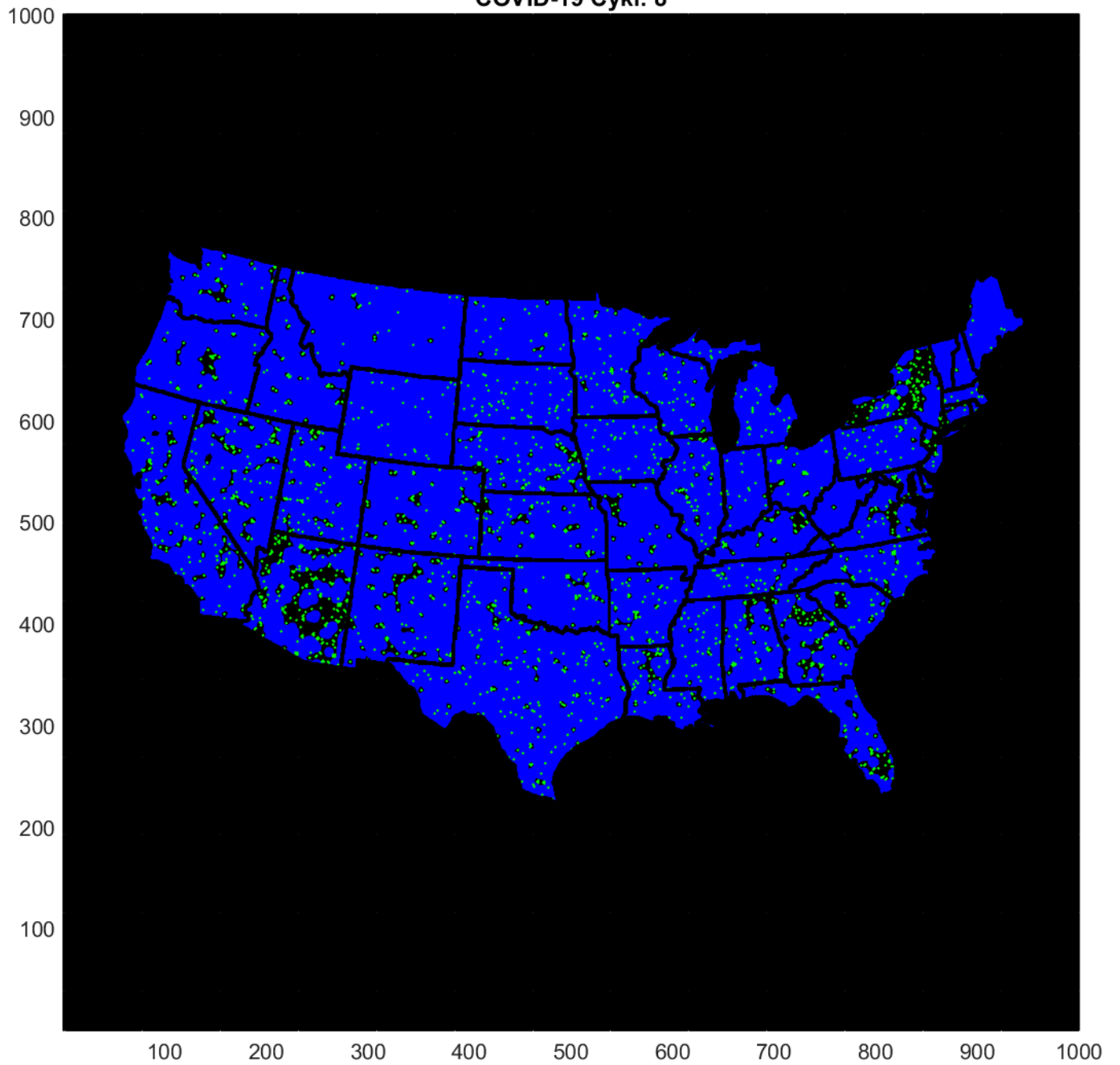
COVID-19 Cykl: 6



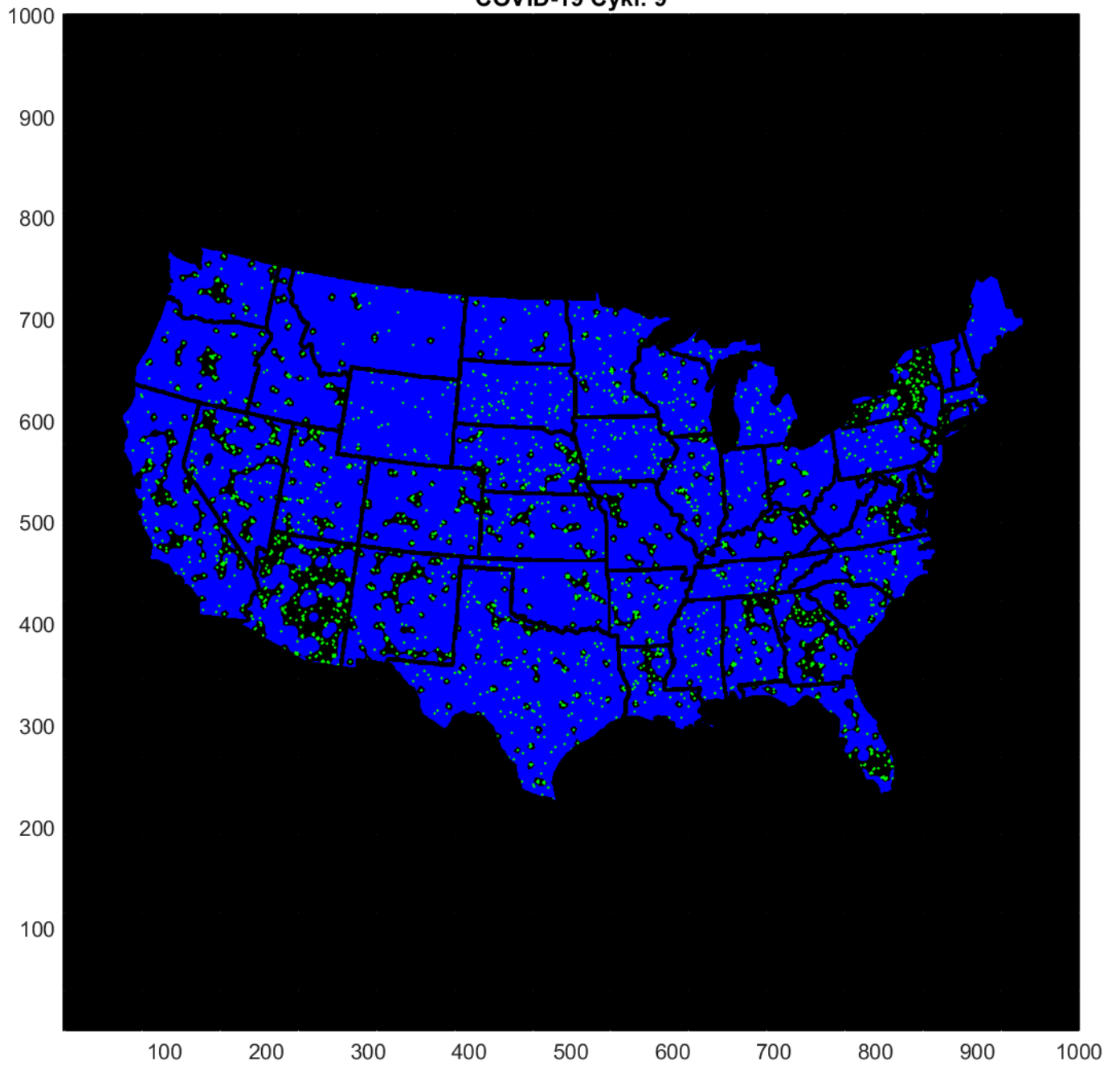
COVID-19 Cykl: 7



COVID-19 Cykl: 8



COVID-19 Cykl: 9



COVID-19 Cykl: 10

