GPS - Problem znajdowania lokalizacji

Jakub Kuczkowiak Grzegorz Ciesielski

20 stycznia 2018

Spis treści

1 Zarys problemu

GPS (Global Positioning System) jest systemem lokalizacji stworzonym przez amerykański departament obrony (Department of Defense) we wczesnych latach '70. Obecnie używa go nie tylko wojsko i służby, ale większość użytkowników smartfonów lub innych urządzeń lokalizacyjnych. Dzięki postępującej komputeryzacji system GPS ogromnie zyskał na popularności, ze względu na swoją dostępność - jedynym wymogiem jest posiadanie urządzenia zdolnego do odbierania sygnałów radiowych na częstotliwościach 1227.60 MHz oraz 1575.42 MHz i wykonywania obliczeń. Głównym zadaniem systemu GPS jest umożliwienie użytkownikom rozpoznawania własnej lokalizacji na powierzchni Ziemi. W poniższej pracy zajmiemy się problemem, jak należy przetworzyć dane otrzymane z satelitów, aby rozpoznać własną lokalizację.

1.1 Szczegóły techniczne problemu

Satelity orbitują Ziemię na relatywnie dużej wysokości (ok. 3-krotności promienia Ziemi) nad powierzchnią. Każda z nich jest wyposażona w bardzo dokładny zegar i komputer obliczający własną pozycję na podstawie trajektorii lotu. Pozycja jest przedstawiana jako punkt w trójwymiarowym układzie kartezjańskim, w którym za punkt $(0, 0, 0)$ przyjęto środek Ziemi, a za jednostkę - jej promień. Ze względów bezpieczeństwa satelity nie odbierają i nie przetwarzają żadnych sygnałów, jedynie wysyłają dane. Sygnał GPS z pojedynczego satelity składa się z 4 składowych - wektora (x, y, z) oznaczającego pozycję satelity, oraz czasu zegara satelity t.
Problem lokalizacji polega na przetworzeniu danych otrzymanych z satelitów tak, aby móc odnaleźć pozycję urządzenia odbierającego sygnał GPS w przyjętym układzie kartezjańskim.

Załóżmy, że znamy odległości d_1, d_2, d_3 między naszym urządzeniem, a pewnymi trzema satelitami. Wówczas naszą pozycję można rozpoznać znając pozycje dowolnych trzech satelitów (x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3) i rozwiązując układ równań kwadratowych:

$$\begin{cases} (x-x_1)^2 + (y-y_1)^2 + (z-z_1)^2 = d_1^2 \\ (x-x_2)^2 + (y-y_2)^2 + (z-z_2)^2 = d_1^2 \\ (x-x_3)^2 + (y-y_3)^2 + (z-z_3)^2 = d_1^2 \end{cases} \tag{1}$$

1.2 Dwa rozwiązania

Powyższy układ równań może mieć jednak dwa rozwiązania (co w rzeczywistości będzie regułą, ponieważ jedno rozwiązanie będzie tylko w niezwykle rzadkim przypadku, gdy nasza pozycja będzie znajdywać się w płaszczyźnie wyznaczonej przez satelity), w związku z czym musimy umieć jedno naturalnie wyeliminować. Jak? Możemy na przykład spojrzeć na długości wektorów obu rozwiązań. Interesuje nas wektor, którego długość jest bliska jeden, gdyż taki punkt będzie leżał blisko powierzchni Ziemi, ponieważ długość wyrażamy w jednostkach promienia Ziemi. Dla obu wyznaczonych rozwiązań obliczamy wartość wyrażenia $|1 - d(x, y, z)| = |1 - \sqrt{x^2 + y^2 + z^2}|$ i wybieramy mniejsze.

1.3 Odległość od satelitów i brak synchronizacji

W rzeczywistości nie znamy jednak odległości od poszczególnych satelitów. Problem ten rozwiązany jest w następujący sposób: każda z nich ma bardzo dokładny zegar zsynchronizowany z pozostałymi satelitami, i poza własną pozycją wysyła odczyt z zegara. Wówczas możemy odmierzyć czas, w którym sygnał z satelity pokonał odległość dzielącą nas od niej i znając średnią prędkość sygnału w atmosferze możemy wyznaczyć odległość od satelity. Jednak tutaj pojawia się problem - nasz zegar urządzenia nie musi być zegarem tak dokładnym, jak zegary umieszczone w satelitach, a jeśli prędkość sygnału jest bliska prędkości światła, to błąd rzędu 10^{-5} sekundy przekłada się na 3000 – metrowy błąd! Dlatego musimy być w stanie zsynchronizować nasz zegar z zegarami satelitów, więc wprowadzamy kolejną niewiadomą: t - błąd naszego zegara. W związku z powyższym potrzebujemy danych z co najmniej 4 satelitów, co pozostawia nam do rozwiązania następujący układ równań:

$$\begin{cases} (x-x_1)^2 + (y-y_1)^2 + (z-z_1)^2 = C^2(t_1 - t)^2 \\ (x-x_2)^2 + (y-y_2)^2 + (z-z_2)^2 = C^2(t_2 - t)^2 \\ (x-x_3)^2 + (y-y_3)^2 + (z-z_3)^2 = C^2(t_3 - t)^2 \\ (x-x_4)^2 + (y-y_4)^2 + (z-z_4)^2 = C^2(t_4 - t)^2 \end{cases} \tag{2}$$

gdzie C jest prędkością sygnału, t_1, t_2, t_3, t_4 - czas podróży sygnału z satelity do urządzenia na podstawie własnego zegara urządzenia.

1.4 Prędkość sygnału

Zauważmy, że w układnie równań (2) założyliśmy tę samą, z góry ustaloną prędkość sygnału dla każdej satelity, a tak wcale nie musi być. Powodem może być różna gęstość atmosfery ziemskiej w zależności od odległości od powierzchni i wysokości geograficznej, na której znajduje się satelita (i w związku z tym zmienna prędkość sygnałów). Jednak w tej pracy nie będziemy się tym zajmować - więcej o możliwych błędach i sposobach ich naprawy można przeczytać w artykule *Global Positioning System: The Mathematics of GPS Receiver* [1] - między innymi można tam przeczytać o systemie PPS (Precise Positioning Service).

1.5 Błąd obliczeń

W większości przypadków dane otrzymane z satelitów będą w pewnym stopniu zaburzone. Jeśli jednak układ (2) ma dwa rozwiązania (co zazwyczaj będzie faktem - patrz sekcja 1.2), niewielkie zaburzenie danych nie zmieni rozwiązywalności układu, jedynie delikatnie zaburzy wynik, w związku z czym nie musimy się przejmować sprzecznością układu (2). W artykule [1] ten problem jest dokładnie opisany.

2 Rozwiązanie problemu

Rozwiązanie problemu sprowadza się do numerycznego rozwiązania układu czterech równań kwadratowych czterech zmiennych (2), lub równoważnie, do znalezienia zera funkcji:

$$F(x,y,z,t) = \begin{bmatrix} f_1(x,y,z,t) \\ f_2(x,y,z,t) \\ f_3(x,y,z,t) \\ f_4(x,y,z,t) \end{bmatrix} = \begin{bmatrix} (x-x_1)^2 + (y-y_1)^2 + (z-z_1)^2 - C^2(t_1-t)^2 \\ (x-x_2)^2 + (y-y_2)^2 + (z-z_2)^2 - C^2(t_2-t)^2 \\ (x-x_3)^2 + (y-y_3)^2 + (z-z_3)^2 - C^2(t_3-t)^2 \\ (x-x_4)^2 + (y-y_4)^2 + (z-z_4)^2 - C^2(t_4-t)^2 \end{bmatrix} \tag{3}$$

2.1 Metoda Newtona - Rhapsona

Do numerycznego znajdywania zer funkcji możemy korzystać z metody Newtona - Rhapsona [2]. Metoda ta, dla funkcji jednej zmiennej, polega na obliczaniu kolejnych przybliżeń zera funkcji poprzez poprowadzenie stycznej do wykresu funkcji w danym punkcie i przyjęcie za kolejne przybliżenie zera otrzymanej stycznej:

$$x_{n+1} = x_n - \frac{f(x)}{f'(x)} \tag{4}$$

Metoda ta wymaga znajomości pochodnej funkcji f.
W przypadku funkcji wielu zmiennych metoda Newtona - Rhapsona ta polega na obliczaniu w kolejnych iteracjach jakobianu J funkcji F i wyznaczaniu kolejnych przybliżeń zera za pomocą wzoru:

$$X_{n+1} = X_n - J^{-1}(X_n) \cdot F(X_n) \tag{5}$$

Dla naszego problemu jakobian J funkcji F w punkcie $X = (x,y,z,t)$ wygląda następująco:

$$J(X) = \begin{bmatrix} \frac{\delta f_1}{\delta x} & \frac{\delta f_1}{\delta y} & \frac{\delta f_1}{\delta z} & \frac{\delta f_1}{\delta t} \\ \frac{\delta f_2}{\delta x} & \frac{\delta f_2}{\delta y} & \frac{\delta f_2}{\delta z} & \frac{\delta f_2}{\delta t} \\ \frac{\delta f_3}{\delta x} & \frac{\delta f_3}{\delta y} & \frac{\delta f_3}{\delta z} & \frac{\delta f_3}{\delta t} \\ \frac{\delta f_4}{\delta x} & \frac{\delta f_4}{\delta y} & \frac{\delta f_4}{\delta z} & \frac{\delta f_4}{\delta t} \end{bmatrix} = \begin{bmatrix} 2x-2x_1 & 2y-2y_1 & 2z-2y_1 & 2C^2t_1 - 2C^2t \\ 2x-2x_2 & 2y-2y_2 & 2z-2y_2 & 2C^2t_2 - 2C^2t \\ 2x-2x_3 & 2y-2y_3 & 2z-2y_3 & 2C^2t_3 - 2C^2t \\ 2x-2x_4 & 2y-2y_4 & 2z-2y_4 & 2C^2t_4 - 2C^2t \end{bmatrix} \tag{6}$$

Jako wektor początkowy dobrze jest przyjąć $X_0 = (0,0,0,0)$ (optymistycznie zakładając zerowy błąd zegara), ponieważ prawidłowe rozwiązanie będzie bliżej punktu $(0,0,0,0)$ niż drugie (patrz: 1.2).

2.2 Metoda algebraiczna

Rozważmy układ równań (2). Po wymnożeniu i przegrupowaniu otrzymujemy:

$$\begin{cases} x^2 + y^2 + z^2 - C^2t^2 + x_1^2 + y_1^2 + z_1^2 - C^2t_1^2 = 2xx_1 + 2yy_1 + 2zz_1 - 2t_1tC^2 \\ x^2 + y^2 + z^2 - C^2t^2 + x_2^2 + y_2^2 + z_2^2 - C^2t_2^2 = 2xx_2 + 2yy_2 + 2zz_2 - 2t_2tC^2 \\ x^2 + y^2 + z^2 - C^2t^2 + x_3^2 + y_3^2 + z_3^2 - C^2t_3^2 = 2xx_3 + 2yy_3 + 2zz_3 - 2t_3tC^2 \\ x^2 + y^2 + z^2 - C^2t^2 + x_4^2 + y_4^2 + z_4^2 - C^2t_4^2 = 2xx_4 + 2yy_4 + 2zz_4 - 2t_4tC^2 \end{cases} \tag{7}$$

Zauważmy następnie, że we wszystkich tych równaniach po lewej stronie występują zmienne x, y, z oraz t w postaci kwadratów. Możemy zatem odjąć pierwsze równanie stronami od każdego z pozostałych, otrzymując w ten sposób układ trzech równań liniowych z czterema niewiadomymi:

$$\begin{cases} 2(x_2-x_1)x + 2(y_2-y_1)y + 2(z_2-z_1)z - 2C^2(t_2-t_1)t = C^2(t_1^2-t_2^2) + x_2^2 + y_2^2 + z_2^2 - (x_1^2+y_1^2+z_1^2) \\ 2(x_3-x_1)x + 2(y_3-y_1)y + 2(z_3-z_1)z - 2C^2(t_3-t_1)t = C^2(t_1^2-t_3^2) + x_3^2 + y_3^2 + z_3^2 - (x_1^2+y_1^2+z_1^2) \\ 2(x_4-x_1)x + 2(y_4-y_1)y + 2(z_4-z_1)z - 2C^2(t_4-t_1)t = C^2(t_1^2-t_4^2) + x_4^2 + y_4^2 + z_4^2 - (x_1^2+y_1^2+z_1^2) \end{cases} \tag{8}$$

Wiemy, że taki układ nie może mieć jednoznacznego rozwiązania. Jeśli jednak dane z satelit są dokładne, to istnieje rozwiązanie wyjściowego układu równań kwadratowych, zatem ten układ liniowy jest zgodny. Można więc wyrazić trzy zmienne x, y oraz z względem parametru t:

$$\begin{cases} x = c_x - t_xt \\ y = c_y - t_yt \\ z = c_z - t_zt \end{cases} \tag{9}$$

dla pewnych stałych c_x, c_y, c_z, t_x, t_y oraz t_z, co odpowiada rozwiązaniu następującego układu macierzy (po przeniesieniu wyrażenia zawierającego t na drugą stronę):

$$\begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) & 2(z_2 - z_1) \\ 2(x_3 - x_1) & 2(y_3 - y_1) & 2(z_3 - z_1) \\ 2(x_4 - x_1) & 2(y_4 - y_1) & 2(z_4 - z_1) \end{bmatrix} \begin{bmatrix} t_x & c_x \\ t_y & c_y \\ t_z & c_z \end{bmatrix} = \\ = \begin{bmatrix} 2C^2(t_2 - t_1) & C^2(t_1^2 - t_2^2) + x_2^2 + y_2^2 + z_2^2 - (x_1^2 + y_1^2 + z_1^2) \\ 2C^2(t_3 - t_1) & C^2(t_1^2 - t_3^2) + x_3^2 + y_3^2 + z_3^2 - (x_1^2 + y_1^2 + z_1^2) \\ 2C^2(t_4 - t_1) & C^2(t_1^2 - t_4^2) + x_4^2 + y_2^2 + z_4^2 - (x_1^2 + y_1^2 + z_1^2) \end{bmatrix} \tag{10}$$

Niech dalej $r_1 = x_1 - c_x$, $r_2 = y_1 - c_y$, $r_3 = z_1 - c_z$. Stosujemy dalej eliminację Gaussa w celu znalezienia tych stałych. Tak obliczone x, y oraz z wystarczy teraz wstawić do dowolnego z pierwotnych czterech równań i pogrupować wyrazy względem potęg t, otrzymując:

$$(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 - C^2(t_1 - t)^2 = 0 \tag{11}$$

$$(t_x t + (x_1 - c_x))^2 + (t_y t + (y_1 - c_y))^2 + (t_z t + (z_1 - c_z))^2 - C^2 t_1^2 + 2C^2 t_1 t - C^2 t^2 = 0 \tag{12}$$

$$t_x^2 t^2 + 2t_x t r_1 + r_1^2 + t_y^2 t^2 + 2t_y t r_2 + r_2^2 + t_z^2 t^2 + 2t_z t r_3 + r_3^2 = 0 \tag{13}$$

$$(t_x^2 + t_y^2 + t_z^2 - C^2)t^2 + 2(C^2 t_1 - t_x r_1 - t_y r_2 - t_z r_3)t + r_1^2 + r_2^2 + r_3^2 = 0 \tag{14}$$

Ostatnie równanie już w jawny sposób podaje nam współczynniki równania kwadratowego:

$$\begin{cases} a = t_x^2 + t_y^2 + t_z^2 - C^2 \\ b = 2(C^2 t_1 - t_x r_1 - t_y r_2 - t_z r_3) \\ c = r_1^2 + r_2^2 + r_3^2 \end{cases} \tag{15}$$

Równanie kwadratowe rozwiązujemy następnie numerycznie poprawnym algorytmem rozwiązywania równań kwadratowych (korzystającym ze wzorów Viete'a). Równanie to powinno mieć dla poprawnych danych dwa rozwiązania, z których jedno musimy odrzucić. Mowa o tym jest w rozdziale 1.2.

3 Więcej satelitów

Przypuśćmy, że mamy dostęp do większej liczby satelitów niż 4 - ze względu na to, że Ziemię orbituje aż 27 satelitów, z dowolnego punktu na Ziemi średnio widać od 8 do 10 z nich. Warto by móc wykorzystać te dodatkowe dane, aby zminimalizować błąd obliczeń. Jak jednak to zrobić?

3.1 Metoda najmniejszych kwadratów

Załóżmy, że mamy dostęp do n satelitów. Wówczas zadanie rozpoznania lokalizacji sprowadza się do rozwiązania układu równań postaci:

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 - C^2(t_1 - t)^2 & = 0 \\ (x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 - C^2(t_2 - t)^2 & = 0 \\ & \vdots \\ (x - x_n)^2 + (y - y_n)^2 + (z - z_n)^2 - C^2(t_n - t)^2 & = 0 \end{cases} \tag{16}$$

Powyższy układ przypomina (2), jednak jest to układ n równań kwadratowych i 4 niewiadomych (gdzie $n > 4$), w związku z czym najprawdopodobniej nie będzie miał rozwiązania. Przyjmijmy zatem oznaczenia:

$$f_i(x, y, z, t) = (x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 - C^2(t - t_i)^2 \tag{17}$$

Spróbujmy zminimalizować kwadrat błędu, tzn. sumę:

$$S(x, y, z, t) = \sum_{i=1}^{n} f_i^2(x, y, z, t) \tag{18}$$

S jest funkcją ciągłą, zatem minimalizuje się w punkcie w którym gradient jest równy zero. Gdyby f była funkcją liniową, można by wprost zastosować metodę najmniejszych kwadratów (bardzo dokładnie opisaną w książce *"Generalized Least Squares"*, patrz: [3]). Jednak f nie jest funkcją liniową i obliczanie gradientu przysparza kłopotów. Zobaczmy zatem:

$$
\begin{aligned}
\frac{\partial S}{\partial x} &= \sum_{i=1}^{n} \frac{\partial}{\partial x} f_i^2 = \sum_{i=1}^{n} 2 \cdot f_i \cdot \frac{\partial f_i}{\partial x} = \\
&= 2 \cdot \sum_{i=1}^{n} \left[(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2 - C^2(t-t_i)^2 \right] (2x - 2x_i) = \\
&= 4 \cdot \sum_{i=1}^{n} \left[(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2 - C^2(t-t_i)^2 \right] (x - x_i)
\end{aligned} \tag{19}
$$

Podobnie w przypadku zmiennych y, z, t. Pozostaje zatem rozwiązać układ:

$$
\frac{\partial S}{\partial x} = \frac{\partial S}{\partial y} = \frac{\partial S}{\partial z} = \frac{\partial S}{\partial t} = 0 \tag{20}
$$

Sprowadza się on do układu czterech równań sześciennych czterech niewiadomych, który mocno komplikuje sprawę, ponieważ liczba rozwiązań się zwiększa i zarazem metoda Newtona - Rhapsona rzadziej przynosi poprawne wyniki.

3.2 Metoda ważona Newtona

We wzorze (5) na kolejne iteracje metody Newtona, możemy zmodyfikować nieco obliczanie odwrotności Jakobianu J^{-1}, to znaczy podstawić pod nią uogólnioną odwrotność: $(J^T W J)^{-1} J^T W$, gdzie J jest pochodną funkcji

$$
F(x, y, z, t) = \begin{bmatrix} f_1(x, y, z, t) \\ f_2(x, y, z, t) \\ \vdots \\ f_n(x, y, z, t) \end{bmatrix} \tag{21}
$$

gdzie f_i jest jak w (17), a W jest dodatnią macierzą wag. W artykule *"An Algebraic Solution of the GPS Equations"* [4] algebraiczny wariant powyższej metody jest bardzo dokładnie przedstawiony.

4 Analiza przykładów

W jaki sposób przeprowadziliśmy testy? Napisaliśmy funkcję o nazwie *gen_test*, która przyjmuje pojedynczy parametr *satellites* oznaczający ilość satelitów, dla których chcemy wygenerować dane. W sposób losowy generuje ona pozycję urządzenia odbierającego sygnał GPS, i następnie losowo dobiera współrzędne satelitów. Metoda ta jest numerycznie poprawna, a dane, które otrzymujemy w jej wyniku są wysokiej precyzji BigFloat. Zwraca ona dwa parametry: X oraz *data*, gdzie data to macierz opisująca dane z satelitów (pozycje + czasy zegara), a X to wektor będący oczekiwanym rozwiązaniem. Jest jednak pewien problem. Funkcja ta zwraca tylko jedno rozwiązanie, podczas gdy istnieją dwa. Zatem w trakcie testowania metod musimy zadbać, aby zwracały one dwa rozwiązania i porównywać z tym, które jest bliżej wygenerowanych danych.

4.1 Metoda algebraiczna

W rozdziale 2.2 przedstawiliśmy dowód poprawności wyprowadzonej metody. Teraz zajmiemy się jej analizą pod względem złożoności i precyzji. Metoda ta jest bardzo szybka, jej złożoność czasowa wynosi $\Theta(1)$, ponieważ musimy jedynie rozwiązać równanie macierzowe stałego rozmiaru 3×3, a następnie wyznaczyć stałe a, b oraz c, dla których rozwiązujemy równanie kwadratowe. Jak dokładna jest ta metoda? Po pierwsze wyznaczamy w niej współczynniki macierzy (10), co może powodować utratę cyfr znaczących w wyniku wykonywanego tam odejmowania jeśli zachodzi: $x_i = x_1$ lub $y_i = y_1$ lub $z_i = z_1$. Oznacza to, że aby doszło do utraty cyfr znaczących, pewna współrzędna urządzenia odbierającego sygnał musi pokrywać się z odpowiednią współrzędną któregoś z satelitów, a taka sytuacja w zasadzie nie ma szans się zdarzyć przy tak dużej dokładności obliczeniowej. Następnie obliczana jest macierz odwrotna poprzez eliminację Gaussa. Metoda ta jest jednak numerycznie poprawna zgodnie ze standardem języka Julia. Pozostaje jeszcze problem wyznaczenia współczynników a, b oraz c równania kwadratowego. Współczynnik c jest zawsze dodatni, więc nie ma z nim problemów. Współczynnik a jest niedokładny jedynie wtedy, gdy suma kwadratów pewnych trzech stałych jest równa kwadratowi prędkości światła. Najbardziej niedokładne może być obliczenie współczynnika b, jednak i tam suma trzech iloczynów musi być równa $C^2 t_1$, co, choć teoretycznie możliwe, nie powinno mieć miejsca w praktyce. Ostatnim etapem jest obliczanie równania kwadratowego. Jednak sprytne zastosowanie wzorów Viete'a pozwala uniknąć utraty cyfr znaczących (zastosowany algorytm jest numerycznie poprawny, gdyż wykonujemy tam jedynie dodawanie i pierwiastkowanie). Podsumowując, do utraty cyfr znaczących dochodzi wtedy, gdy:

$$x_i = x_1 \text{ lub } y_i = y_1 \text{ lub } z_i = z_1 \text{ lub } x_i^2+y_i^2+z_i^2 = x_1^2+y_1^2+z_1^2 \text{ lub } t_x^2+t_y^2+t_z^2 = C^2 \text{ lub } C^2 t_1 = t_x r_1+t_y r_2+t_z r_3 \tag{22}$$

Obliczenia wykonujemy na typie danych BigFloat, co pozwala na zachowanie ogromnej precyzji obliczeń. Jak dokładność wpływa na nasz wynik? Załóżmy, że obliczyliśmy współrzędną w jednostkach promienia Ziemi, na której się znajdujemy z dokładnością do k miejsc po przecinku. Promień Ziemi wynosi w przybliżeniu $6.4 * 10^8$ m. Oznacza to, przy założeniu, że otrzymaliśmy dokładne dane z satelitów, że błąd naszej pozycji w metrach ma dokładność do $k - 9$ miejsc po przecinku. Dla przykładu, rozważamy obliczanie następującego równania:

$$\begin{cases} (x-234)^2 + (y-123)^2 + (z-342)^2 = (354-t)^2 \\ (x-231)^2 + (y-234)^2 + (z-413)^2 = (432-t)^2 \\ (x-321)^2 + (y-352)^2 + (z-643)^2 = (642-t)^2 \\ (x-213)^2 + (y-302)^2 + (z-521)^2 = (523-t)^2 \end{cases} \tag{23}$$

Porównanie wyników z naszego programu w precyzji BigFloat 256-bitowej i wyników z WolframAlpha (nasze wyżej):

$$\begin{cases} x = 223.97605295001951333752434779788694572788843073598178637064134221807226910692 \\ x = 223.97605295001951333752434779788694572788843073598178637064134221807226910690 \\ y = 61.048557303396345957398534770706925260983886323820398949351649398096532308487 \\ y = 61.048557303396345957398534770706925260983886323820398949351649398096532308483 \\ z = 413.44592219605926701089190456163589205052548031470293130169904238986693929887 \\ z = 413.44592219605926701089190456163589205052548031470293130169904238986693929885 \\ t = 258.90541253521979255169217359014046413298532297538496213700950084891090986 08 \\ t = 258.9054125352197925516921735901404641329853229753849621370095008489109098607 \end{cases} \tag{24}$$

Co dla podanego przykładu oznacza błąd rzędu 10^{-73}, czyli pomyłkę o około 10^{-64} metra.

Metoda algebraiczna wyraża się dokładnym wzorem i wysoką precyzją. Jej uwarunkowanie jest, poza bardzo rzadkimi przypadkami, bardzo dobre. Aby przekonać się, jak dobra jest to metoda, postanowiliśmy napisać funkcję *test*, która wykorzystuje omawianą wcześniej funkcję *gen_test*. Wygenerowaliśmy w ten sposób ogromną ilość danych i poddaliśmy je próbie naszej metody, sprawdzając dokładność błędu na precyzji 256-bitowej BigFloat. Nasza metoda na 100 000 000 losowo wygenerowanych testach miała największy błąd rzędu 10^{-54}.

4.2 Metody Newtona

Metoda Newtona - Rhapsona jest zbieżna kwadratowo, w związku z czym liczba iteracji wymaganych do znalezienia zera funkcji F jest niewielka (w większości testów nie przekracza ona 20, przy obliczeniach w arytmetyce 256 bitowej z dokładnością maszynową).
Problemem tej metody jest fakt, iż znajduje ona jedynie jedno rozwiązanie z potencjalnych dwóch. Rozwiązaniem może być losowanie wektora początkowego - oczekiwana liczba wymaganych losowań wektora początkowego do trafienia wszystkich rozwiązań jest równa $2 \ln 2$ (patrz: problem kolekcjonera [5]).

Obie metody Newtona (zwykła, dla 4 satelitów i ważona, dla większej ich ilości) uzyskiwały bardzo dobre wyniki. Przykładowo, klasyczna metoda Newtona – Rhapsona dla 100 000 losowo wygenerowanych testów osiągnęła średnią dokładność do 76 miejsc po przecinku, a największy błąd był do 72 miejsca po przecinku, a metoda ważona Newtona dla 100 000 testów z 10 satelitami osiągnęła średnią dokładność do 78 miejsc po przecinku, a największy błąd był do 76 miejsca po przecinku. Za cenę złożoności otrzymujemy wyniki dokładniejsze, niż w metodzie algebraicznej.

4.3 Metoda najmniejszych kwadratów

Ze względu na wymieniony wcześniej fakt, że funkcja (17) nie jest liniowa, minimalizacja sumy kwadratów wartości funkcji nie jest prosta przysparza wielu kłopotów obliczeniowych. W testach otrzymana zbieżność metody jest bardzo kiepska, i bardzo często metoda nie jest zbieżna do minimum globalnego, lecz lokalnego, w związku z czym często nie daje poprawnych wyników i metoda Newtona – Rhapsona jest pod każdym względem lepsza.

5 Podsumowanie

Z powyższych wniosków wynika, że metoda algebraiczna daje znakomite wyniki i że błędów obliczeń nie odczujemy w praktyce. Wniosek ten specjalnie nie dziwi - mamy bowiem konkretny wzór, w którym wykonujemy jedynie podstawowe operacje arytmetyczne, a uwarunkowanie zadania jest zdecydowanie dobre - przypadki patologiczne, jeśli nawet się zdarzą, nie wpłyną istotnie na rozwiązanie. Inną sprawą jest to, jak bardzo dokładne dane dostaniemy i czy podczas transferu danych nie dojdzie do utraty bitów - tego problemu w tej pracy już nie poruszamy i zostawiamy do rozwiązania inżynierom (patrz: [1]). Metoda algebraiczna świetnie nadaje się do czterech satelitów, jednak ich większa ilość nie poprawia dokładności. Sposób algebraicznego rozwiązania wykorzystujący dane z większej ilości satelitów można zobaczyć w artykule [4]. Do naszych testów wykorzystaliśmy metodę ważoną Newtona, która spełniała swoje zadanie - faktycznie wyniki były dokładniejsze, niż w przypadku metod korzystających z mniejszej liczby danych. Co jednak z najbardziej klasycznym podejściem znajdowania zera funkcji za pomocą metody Newtona-Rhapsona? Pomimo, że stosunkowa mała liczba iteracji daje wystarczającą dla nas precyzję, to ma ona zasadniczą wadę - aby wyznaczyć oba rozwiązania i znaleźć to właściwe, musimy wprowadzić do niej randomizację wektora początkowego, co znacznie może wydłużyć czas potrzebny na obliczenie pozycji. To z kolei prowadzi do szybszego wyczerpania baterii, co w przypadku tego typu urządzeń bywa niezwykle uciążliwe i z pewnością każdy z nas doświadczył tego problemu. Urządzenia te bowiem przetwarzają pozycję cały czas, w związku z czym ważna jest dla nas nie tylko precyzja obliczeń, ale również ich szybkość.

Literatura

[1] Richard M. Thompson. Global Positioning System: The Mathematics of GPS Receivers. *Mathematics Magazine*, 71(4):260–269, 1998.

[2] Adi Ben-Israel. A Newton-Raphson Method for the Solution of Systems of Equations. *Journal of Mathematical Analysis and Applications*, 15(2):243–252, 1966.

[3] Takeaki Kariya and Hiroshi Kurata. *Generalized Least Squares*. 2004.

[4] Stephen Bancroft. An algebraic solution of the GPS equations. *IEEE Transactions on Aerospace and Electronic Systems*, (1):56–59, 1985.

[5] Arnon Boneh and Micha Hofri. The coupon-collector problem revisited—a survey of engineering problems and computational methods. *Stochastic Models*, 13(1):39–66, 1997.