

# Tache6

---

Pascal Isak && Weber Loïc

## Table Of Content

---

- [fonction distance\\_point\\_segment](#)
- [test\\_simplification\\_contour](#)

## fonction distance\_point\_segment

---

geometrice.c

```
double distance_point_segment(Point P, Point A, Point B){
    if (A.x == B.x && A.y == B.y){
        return distance_point(A,P);
    }
    // On est dans le cas A != B :

    // On commence par calculer lambda
    double lambda = produit_scalaire(couple_point_to_vecteur(A,P),couple_point_to_vecteur(A,B));

    // Cas lambda < 0 :
    if (lambda < 0){
        return distance_point(A,P);
    }

    // Cas lambda > 1 :
    if (lambda > 1){
        return distance_point(B,P);
    }

    // Dernier cas 0 <= lambda <= 1 :
    // On calcule le point Q :
    Point Q = addition_point(A, produit_point(lambda, addition_point(B, negation_point(A,B))));
    //Point Q = nouveau_point(A.x + lambda * (B.x - A.x), A.y + lambda * (B.y - A.y));
    return distance_point(Q,P);
}
```

## test\_simplification\_contour

---

test\_simplification\_contour

```

void jeu_de_test(){

    printf("Test simplification contour\n");
    printf("Test 1/10 : ");
    afficher_resultat_test(distance_point_segment(nouveau_point(0,0), nouveau_point(0,0)));

    printf("Test 2/10 : ");
    afficher_resultat_test(distance_point_segment(nouveau_point(0.5,0), nouveau_point(0.5,0)));

    printf("Test 3/10 : ");
    afficher_resultat_test(distance_point_segment(nouveau_point(3.5,3.5), nouveau_point(3.5,3.5)));

    printf("Test 4/10 : ");
    afficher_resultat_test(distance_point_segment(nouveau_point(5,8), nouveau_point(5,8)));

    printf("Test 5/10 : ");
    afficher_resultat_test(distance_point_segment(nouveau_point(5,8), nouveau_point(5,8)));

    printf("Test 6/10 : ");
    afficher_resultat_test(distance_point_segment(nouveau_point(5,11), nouveau_point(5,11)));

    printf("Test 7/10 : ");
    afficher_resultat_test(distance_point_segment(nouveau_point(5,-1), nouveau_point(5,-1)));

    printf("Test 8/10 : ");
    afficher_resultat_test(distance_point_segment(nouveau_point(7,8), nouveau_point(7,8)));

    printf("Test 9/10 : ");
    afficher_resultat_test(distance_point_segment(nouveau_point(8,14), nouveau_point(8,14)));

    printf("Test 10/10 : ");
    afficher_resultat_test(distance_point_segment(nouveau_point(1,2), nouveau_point(1,2)));
}

int main(int argc, char** argv){
    jeu_de_test();

    // Test en ligne de commande :
    if (argc != 7){
        printf("Usage: ./test_simplification_contour P.x P.y A.x A.y B.x B.y\n");
        return 1;
    }
    Point P = nouveau_point(atof(argv[1]),atof(argv[2]));
    Point A = nouveau_point(atof(argv[3]),atof(argv[4]));
    Point B = nouveau_point(atof(argv[5]),atof(argv[6]));

    double resultat = distance_point_segment(P, A, B);
    printf("distance_point_segment = %f \n",resultat);
    return 0;
}

```