Numerical computations

Derumigny, A. , Guyonvarch Y. & Girard, L.

September 8, 2022

```
library("cubature")
library("expint")
```

Optimization of the components of Ω_1

Bound on $I_{1,1}$

```
# First term of Omega_1
toIntegrate_1_1 <- function(t, capitalT) {
  indic = as.numeric(t <= 1)
  module = sqrt( (1 - t)^2 * indic +
                   ( ( (1-t) * cos(pi * t) / sin(pi * t) + 1/pi) * indic
                     - 1/(pi * t) )^2 )
  expo = exp( - (t * capitalT)^2 / 2)

  return (module * expo)
}

# Function to optimize
Integral_1_1 <- function (capitalT){
  integral = -hcubature(f = toIntegrate_1_1, lower = 0, upper = 1/pi,
                        capitalT = abs(capitalT))$integral
  return(as.numeric(capitalT * integral))
}

# Computation of the maximum of I_{1,1}
result_1_1 =
  optim(f = Integral_1_1, par = 1,
        method = "L-BFGS-B", control = list(trace = 1))
```

```
## iter   10 value -1.236798
## iter   20 value -1.252322
## iter   30 value -1.253254
## final  value -1.253279
## converged
```

```
print(- result_1_1$value)
```

```
## [1] 1.253279
```

Bound on $I_{1,2}$

```
# Second term of Omega_1
toIntegrate_1_2 <- function(t, capitalT) {
  indic = as.numeric(t <= 1)
  module = sqrt((1 - t)^2 * indic +
                  ( ( (1-t) * cos(pi * t) / sin(pi * t) + 1/pi) * indic
                    - 1/(pi * t) )^2 )
  expo = exp( - (t * capitalT)^2 / 2) * t^3 / 6

  return (module * expo)
}

# Function to optimize
Integral_1_2 <- function (capitalT){
  integral = -hcubature(f = toIntegrate_1_2, lower = 0, upper = 1/pi,
                        capitalT = abs(capitalT))$integral
  return(as.numeric(capitalT^4 * integral))
}

# Computation of the maximum of I_{1,2}
result_1_2 =
  optim(f = Integral_1_2, par = 1,
        method = "L-BFGS-B", control = list(trace = 1))
```

```
## iter   10 value -0.332670
## iter   20 value -0.333293
## final  value -0.333311
## converged
```

```
print(- result_1_2$value)
```

```
## [1] 0.3333106
```

Bound on $I_{1,3}$

```
Integral_1_3 <- function (capitalT){
  value = - capitalT^4 * gammainc(0, capitalT^2 / (2 * pi^2) ) / (2*pi)
  return(value)
}

# Computation of the maximum of I_{1,3}
result_1_3 =
  optim(f = Integral_1_3, par = 1,
        method = "L-BFGS-B", control = list(trace = 1))
```

```
## final  value -14.196144
## converged
```

```
print(- result_1_3$value)
```

```
## [1] 14.19614
```

Bound on $I_{1,4}$

```
Integral_1_4 <- function (capitalT){
  value = - capitalT^3 * gammainc(3/2, capitalT^2 / (2 * pi^2) ) / (3*sqrt(2)*pi)
  return(value)
}

# Computation of the maximum of I_{1,3}
result_1_4 =
  optim(f = Integral_1_4, par = 1,
        method = "L-BFGS-B", control = list(trace = 1))
```

```
## final  value -4.339341
## converged
```

```
print(- result_1_4$value)
```

```
## [1] 4.339341
```

Optimization of the components of Ω_2

Computation of t_1^* and χ_1

```
t0 = 1/pi

result_t1 =
  optim(
    fn = function(theta){return(abs(
      theta^2 + 2 * theta * sin(theta) + 6 * (cos(theta) - 1 ) ))},
    lower = 3.8, upper = 4.2, par = 4,
    method = "L-BFGS-B", control = list(trace = 1))
```

```
## final  value 0.000005
## stopped after 3 iterations
```

```
t1star = result_t1$par / (2*pi)
print(t1star)
```

```
## [1] 0.6359664
```

```
result_chi1 =
  optim(f = function(x){return( - abs(cos(x)-1 + x^2/2) / x^3)},
        par = 1, lower = 10^(-5),
        method = "L-BFGS-B")
chi1 = - result_chi1$value
print(chi1)
```

```
## [1] 0.09916191
```

Bound on $I_{2,1}$

```
toIntegrate_2_1 <- function(t, capitalT) {
  indic = as.numeric(t <= 1)
  module = sqrt((1 - t)^2 * indic +
                  ( ( (1-t) * cos(pi * t) / sin(pi * t) + 1/pi)
                * indic )^2 )
```

```
  expo = exp( - 0.5 * (t * capitalT)^2 * ( 1 - 4*pi*chi1*abs(t) ) )
  return (module * expo)
}

# Function to optimize
Integral_2_1 <- function (capitalT){
  integral = - hcubature(f = toIntegrate_2_1, lower = t0, upper = t1star,
                         capitalT = abs(capitalT))$integral
  return(as.numeric(capitalT^4 * integral))
}

# Computation of the maximum of I_{2,1}
result_2_1 =
  optim(f = Integral_2_1, par = 1, lower = 0,
        method = "L-BFGS-B", control = list(trace = 1))
```

```
## final  value -67.041372
## converged
```

```
print(- result_2_1$value)
```

```
## [1] 67.04137
```

Bound on $I_{2,2}$

```
toIntegrate_2_2 <- function(t, capitalT) {
  indic = as.numeric(t <= 1)
  module = sqrt((1 - t)^2 * indic +
                  ( ( (1-t) * cos(pi * t) / sin(pi * t) + 1/pi)
                * indic )^2 )
  expo = exp( - capitalT^2 * ( 1 - cos(2*pi*t) ) / (4 * pi^2) )
  return (module * expo)
}

# Function to optimize
Integral_2_2 <- function (capitalT){
  integral = - hcubature(f = toIntegrate_2_2, lower = t1star, upper = 1,
                         capitalT = abs(capitalT))$integral
  return(as.numeric(capitalT^2 * integral))
}

# Computation of the maximum of I_{2,2}
result_2_2 =
  optim(f = Integral_2_2, par = 1, lower = 0,
        method = "L-BFGS-B", control = list(trace = 1))
```

```
## final  value -1.218606
## converged
```

```
print(- result_2_2$value)
```

```
## [1] 1.218606
```