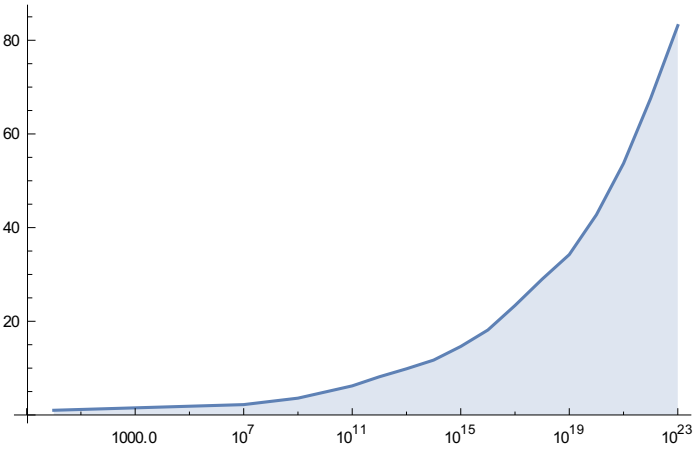
In[6]:=
```
(* List of fast Deleglise-Rivat alpha factors found by running pi(x) benchmarks. *)

alpha = {(* {x, alpha} *){1, 1}, {10^7, 2.210}, {10^9, 3.590},
  {10^10, 4.900}, {10^11, 6.211}, {10^12, 8.154}, {10^13, 9.857}, {10^14, 11.715},
  {10^15, 14.628}, {10^16, 18.138}, {10^17, 23.375}, {10^18, 28.992},
  {10^19, 34.240}, {10^20, 42.741}, {10^21, 53.706}, {10^22, 67.605}, {10^23, 83.126}}
```

Out[6]=
```
{{1, 1}, {10 000 000, 2.21}, {1 000 000 000, 3.59}, {10 000 000 000, 4.9}, {100 000 000 000, 6.211},
 {1 000 000 000 000, 8.154}, {10 000 000 000 000, 9.857}, {100 000 000 000 000, 11.715},
 {1 000 000 000 000 000, 14.628}, {10 000 000 000 000 000, 18.138}, {100 000 000 000 000 000, 23.375},
 {1 000 000 000 000 000 000, 28.992}, {10 000 000 000 000 000 000, 34.24},
 {100 000 000 000 000 000 000, 42.741}, {1 000 000 000 000 000 000 000, 53.706},
 {10 000 000 000 000 000 000 000, 67.605}, {100 000 000 000 000 000 000 000, 83.126}}
```

In[7]:=
```
ListLogLinearPlot[alpha, Filling → Bottom, Joined → True]
```

Out[7]=




```
(* alpha is a tuning factor that balances the compuation of the easy special
 leaves and the hard special leaves. The formula below is used in the file
 src/common.cpp to calculate a fast alpha factor for the computation of pi(x). *)
```

In[8]:=
```
NonlinearModelFit[alpha, a (Log[x])^3 + b (Log[x])^2 + c Log[x] + d, {a, b, c, d}, x]
```

Out[8]=
```
FittedModel[ 0.372253 + 1.00165 Log[x] - 0.0691909 ≪1≫^2 + 0.00148918 Log[x]^3 ]
```