```
(* Step 1: Shannon Channel Capacity *)
C[B_, S_, N_] := B Log2[1 + S/N] (* Equation [1] *)


(* Step 2: Hoyt Distribution PDF *)
f[x_, q_, m_] := 2 q (q/m)^q x^(2 q - 1) Exp[-(q/m) x^2] (* Equation [2] *)
(* Step 3: PDF of Gaussian Noise *)
f_noise[x_, σ_] := 1/(Sqrt[2 π] σ) Exp[-x^2/(2 σ^2)]


(* Step 4: MIMO Channel Capacity with Noise *)
C_with_noise[B_, H_, x_, P_, σ_] :=
 Expectation[Log2[Det[I + H.x.P.ConjugateTranspose[H] + σ^2 IdentityMatrix[NR]]],
  Distributed[x, MultinormalDistribution[{0, 0, ...}, P]]] (* Equation [6] *)


(* Step 5: Matrix Determinant Lemma Simplification *)
det_simplified[x_, H_, P_, σ_] :=
 Det[x.P.ConjugateTranspose[H].H.x + σ^2 IdentityMatrix[NR]]/Det[P] (* Equation [2] *)


(* Step 6: Logarithm and Expectation Simplification *)
log_det_expectation[x_, H_, P_, σ_] :=
 Log[Det[P]] +
 Expectation[Log[Det[I + ρ^(-1/2) W]],
  Distributed[x, MultinormalDistribution[{0, 0, ...}, P]]] (* Equation [3] *)


(* Step 7: Expectation Approximation using Law of Large Numbers *)
E_log_det_W_approx[q_, m_, ρ_] := NR Expectation[Log[1 + ρ^(-1/2) λi], λi \[Distributed] W] (*
Equation [9] *)


(* Step 8: Using Hoyt Distribution for Eigenvalues *)
E_log_det_W_hoyt[q_, m_, ρ_] :=
 NIntegrate[Log[1 + ρ^(-1/2) λ] f_approx[λ, q, m], {λ, 0, ∞}] (* Equation [10] *)
(* Step 9: Low SNR Approximation of Hoyt Distribution PDF *)
```

```
f_approx[x_, q_, m_] := (2 q/m)^(q + 1) x^(2 q) Exp[-(2 q/m) x^2] (* Equation [3] *)

(* Step 10: Substituting Hoyt Distribution into Expectation *)

E_log_det_W_hoyt_sub[q_, m_, ρ_] :=

  (2 q/m)^(q + 1) NIntegrate[Log[1 + ρ^(-1/2) λ] λ^(2 q) Exp[-(2 q/m) λ], {λ, 0, ∞}] (* Equation [11] *)


(* Step 11: Change of Variable in Equation [11] *)

E_log_det_W_hoyt_sub_t[q_, m_, ρ_] :=

  (2 q/m)^(q + 1) ρ^(q + 1/2) NIntegrate[Log[1 + t] t^(2 q) Exp[-(2 q/m) ρ t^2], {t, 0, ∞}] (* Equation [12] *)

(* Step 12: MIMO Channel Capacity *)

(* Assuming matrices H, x, and n are given *)

y = H.x + n; (* Equation [4] *)


(* Step 13: Low SNR Approximation of Hoyt Distribution for Eigenvalues *)

(* Assuming matrix W is given *)

E_log_det_W[q_, m_, ρ_] := NIntegrate[Log[1 + ρ^(-1/2) λ] f_approx[λ, q, m], {λ, 0, ∞}] (* Equation [6] *)


(* Step 14(a): Water-filling Algorithm for Power Allocation *)

(* Assuming constraints and parameters are given *)

(* Note: This is a simplified representation, and actual implementation may involve optimization solvers *)

P_optimal = WaterFillingAlgorithm[E_log_det_W[q, m, ρ], Tr[P] <= Pmax] (* Equation [13] and [14] *)


(* Step 14(b): MIMO Channel Capacity in Low SNR Regime *)

(* Assuming parameters are given *)

λmax = Sqrt[NR/(2 q)] (1 + Sqrt[(2 q/m) NR/SNR])^2; (* Equation [27] *)

C_lsnr = NT (Log2[1 + SNR/NR λmax] - Log2[E] (2 q (NT - NR)/(NR λmax))) (* Equation [26] *)


(* Additional Steps: Incorporating Fading and Noise Effects *)

(* Assuming additional parameters and formulas are given *)
```

```
(* Step 15: Optimal Power Allocation using Lagrange Multipliers *)

(* Note: This is a simplified representation; actual implementation may involve optimization solvers *)

lagrange_multiplier[P_, λ_] := E_log_det_W_hoyt_sub[q, m, ρ] + λ (Tr[P] - Pmax)


(* Step 16: Water-filling Algorithm *)

(* Note: This is a simplified representation; actual implementation may involve optimization solvers *)

water_filling_algorithm[P_] := Solve[D[lagrange_multiplier[P, λ], P] == 0, P]

(* Steps 17-23: Remaining steps involving the final capacity expression *)

(* Assuming parameters are given *)


(* Define SNR in terms of the given variables *)

SNR_expr[P_] := Total[P]/σ^2


(* Define λmax and λ in terms of the given variables *)

λmax_expr[SNR_] := Sqrt[NR/(2 q)] (1 + Sqrt[(2 q/m) NR/SNR])^2

λ_expr[SNR_] := (NR/(2 q)) (1 + Sqrt[(2 q/m) NR/(Total[P]/σ^2)])^2


(* Step 24: Incorporating SNR and Fading into Capacity Equation *)

C_final[SNR_, q_, m_, P_, σ2_] :=
  NT Log2[1 + Total[P]/(σ2 NR) λmax] - Log2[E] (2 q (NT - NR)/(NR λmax))


(* Plotting the MIMO channel capacity *)

ListLinePlot[Transpose[{SNR_dB, C_lsnr}], Mesh -> All,
  Frame -> True, FrameLabel -> {"SNR (dB)", "Capacity (bits/s/Hz)"},
  PlotLabel -> "MIMO channel capacity in low SNR",
  PlotLegends -> {"Hoyt fading"}]
```