Curso Latex + Mathematica (2024)

In[ ]:=

Impetus (revision) Isothermal Bondi accretion in Galaxies with a Central Black Hole: Fully Analytical Solutions

Continuation of the IMPETUS Project by: JM. Ramirez-Velasquez @ YachayTech 2024.

Es importante mencionar que los departure de las estimaciones con respecto al proceso de Bondi por ejemplo, es una aplicacion prioritaria en estos estudios.

Los estudios observacionales no tienen la resolucion para resolver las escalas de parsec! .. Asi que la gente se agarra de Bondi. Pero no queda bien, porque es overestimated o underestimated.

Unfortunately, in general, these studies lack the resolution to follow gas transport down to the parsec scale, and the Bondi model is used as the starting point for estimates of the accretion radius (i.e., the sonic radius) and of the mass accretion rate. In particular, the Bondi accretion rate gives the mass supply to the MBH by taking the density and temperature at some finite distance from the center, implicitly assuming that these values represent the true boundary conditions (i.e., at infinity) for the Bondi problem.

```
nbook = EvaluationNotebook[];
tA = SessionTime[];
myDir = NotebookDirectory[];
(**-- Corro celda por celda --**)
Do[
 SelectionMove[nbook, Next, Cell, 1];
 SelectionEvaluate[nbook];
 , {i, 1, 124}]
```

The Classical Bondi Model

The polytropic gas sound speed is

$$c_{s2} = \gamma \frac{p}{\rho} \quad (*2*)$$

The isothermal case is recovered for $\gamma = 1$. The time-independent continuity equation is (spherically symmetric)

$$\dot{M}_B = 4\pi r^2 \rho[r] \times v[r] \quad (*3*)$$

Bueno con unas normalizacion etc, la ecuacion de continuidad queda en una forma muy compacta:

$$Eq8 = x^2 \mathcal{M} \tilde{\rho}^{\frac{\gamma+1}{2}} == \lambda \quad (*8*)$$

$$\text{Inactive}\left[\lambda = \frac{\dot{M}_B}{4\pi r_B^2 \rho_\infty c_\infty}\right]$$

Supongo que podemos escribir Bernoulli normalizado

$$Eq9a = \frac{\mathcal{M}^2 \tilde{c}_s{}^2}{2} + \frac{\tilde{\rho}^{\gamma-1}}{\gamma - 1} == \frac{1}{x} + \frac{1}{\gamma - 1} \quad (*9a*)$$

Por ejemplo, puedo despejar yo c_s? (Respuesta!!.. si)

```
Solve[Eq9a, c̃s]
```

***λ* is the dimensionless accretion parameter that determines the accretion rate for assigned *M_{BH}* and boundary conditions**. From Equations (7)–(8), the radial profile of all hydrodynamical properties can be expressed in terms of (x). By elimination of r˜ in Equations (8)–(9) for 1≤

γ≤5/3, the Bondi problem reduces to the solution of the equation

Y por ejemplo, puedo despejar yo $\mathcal{M}$? (Respuesta!!.. si)

```
In[ ]:= SolM1 = Solve[{Eq8, Eq9a}, {M, x}]
```

Para que la solucion exista:

```
In[ ]:= Λ ≥ gmin/fmin (*condition*)
```

```
In[ ]:= λcr = (1 / 4) (2 / (5 - 3 γ))^((5-3 γ) / (2 (γ-1)))  (* 15 *)
```

```
In[ ]:= λcr /. γ → 1.4
```

```
In[ ]:= M /. SolM1[[1]];
```

```
In[ ]:= γ1 = 1.4;
M /. SolM1[[2]] /. {ρ̃ → 1, c̃s → 2, γ → γ1, λ → 0.01}
```

```
In[ ]:= EqMio1 = ((Solve[Eq9a, x] /. {ρ̃ → 1, c̃s → 1, γ → 1.7}))[[1]][[1]][[2]]
```

```
In[ ]:= LogLogPlot[EqMio1, {M, 0, 3}]
```

```
In[ ]:= EqMio2 = (Solve[Eq8, M][[1]][[1]][[2]] /. {ρ̃ → 1, λ → λcr}) /. γ → 1.1
```

```
In[ ]:= lineStyle = {Thick, Red, Dashed};
line1 = Line[{{1, 0}, {1, 10}}];
test1 = Plot[{1, EqMio2}, {x, 0.6, 2}, Epilog → {line1}]
```

Deberia hacer (11) tambien:

```
In[ ]:= gGeneral[M] = M^(2 (1-γ) / (γ+1)) (M^2/2 + 1/(γ - 1)) (*11*)
```

```
In[ ]:= Mgmin1 =
  Solve[D[gGeneral[M], {M, 1}] == 0, M]
```

```
In[ ]:= Mgmin2 = gGeneral[M] /. Mgmin1[[1]] (*13*)
```

```
In[ ]:= fGeneral[x] = x^(4 (1-γ) / (γ+1)) (1/x + 1/(γ - 1)) (*12*)
```

```
In[ ]:= Mfmin1 =
  Solve[D[fGeneral[x], {x, 1}] == 0, x]
```

```
In[ ]:= Mfmin2 = fGeneral[x] /. Mfmin1[[1]] (*14*)
```

La condicion empieza aqui:

```
In[ ]:= Reduce[Mfmin1[[1]][[1]][[2]] ≥ 0, γ]
```

Por ejemplo:

```
In[ ]:= Mfmin1[[1]][[1]][[2]] /. γ → 3 / 5 - 0.1
In[ ]:= Mfmin2 /. γ → 3 / 5 - 0.1
        Mgmin2 /. γ → 3 / 5 - 0.1
In[ ]:= Λmin = Mgmin2 / Mfmin2 // FullSimplify
In[ ]:= Λmin[[3]]
```

Veo desde donde deberia valer λ

```
In[ ]:= NSolve[Λmin[[2]] > 0, γ]
```

Estoy revisando los detalles de la formulacion en terminos de λ. Lo primero que queria chequear es que $\lambda_{cr} = e^{3/2} / 4$ for $\gamma \to 1^{+}$.

```
In[ ]:= λcr = (1 / 4) (2 / (5 - 3 γ))^((5-3 γ) / (2 (γ-1)))  (* 14 *)
In[ ]:= test2 = Plot[{λcr, e^(3/2) / 4, 1 / 4}, {γ, 0, 1.65},
          PlotRange → {{0, 2}, {0, 3}}, PlotStyle → Thick]
In[ ]:= N[5 / 3]
In[ ]:= λcr /. {γ → 1.666}
In[ ]:= Limit[λcr, γ → 1]
In[ ]:= Limit[λcr, γ → 5 / 3]
```

In the isothermal case,

```
In[ ]:= Eq15 = f[x] == g[M] + Log[λ] (* 15 *)
In[ ]:= g[M] = M^2 / 2 - Log[M] (*16a*)
In[ ]:= Solve[D[g[M], {M, 1}] == 0, M]
In[ ]:= gmin = g[M] /. M → 1
In[ ]:= test2 = Plot[Evaluate[g[M]], {M, 0, 1.6},
          Epilog → {PointSize[Large], Point[{1, gmin}]},
          PlotRange → {{0, 1.6}, {0, 2.2}}]
In[ ]:= f[x] = 1 / x + 2 Log[x] (*16b*)
In[ ]:= Solve[D[f[x], {x, 1}] == 0, x]
In[ ]:= fmin = f[x] /. x → 1 / 2
```

```
test3 = Plot[Evaluate[f[x]], {x, 0, 1},
  Epilog → {PointSize[Large], Point[{1/2, f_min}]},
  PlotRange → {{0, 1.4}, {0, 5}}]
```

Bueno, todo (17) es correcto!

Esta es una condicion impuesta por (10)!.. Osea λ tiene (para el caso isitermico) que ser ≤ que λ_{crIso}

```
λ_crIso = e^(f_min - g_min)
```

No entran en detalles de las soluciones numericas. Solo se concentran en $\lambda = \lambda_{cr}$, donde x_{min} es el punto sonico. De las dos soluciones criticas, consideramos solo la que hace que $\mathcal{M}$ aumente a medida que se aproxime al centro.

Mass accretion bias: concepts

For assigned values of ρ_∞, T_∞, γ and M_{BH} the classical Bondi accretion rate is given by:
(super importante!! esa λ es realmente λ_{cr}!! asi que ese Bondi rate, es calculado con esa lambda)

$$Eq18 = \dot{M}_B == 4 \pi r_B^2 \lambda \rho_\infty c_\infty \quad (* \ 18 \ *)$$

In practice, when dealing with observations or numerical simulations, one inserts in eq. (18) the values of ρ and T at a **finite distance** r from the MBH, and considers them as “proxies” for ρ_∞ and T_∞. This procedure gives an estimated value of the Bondi radius (that we call r_e) and mass accretion rate (that we call $\dot{M}_e$). Here we investigate how much these r_e and $\dot{M}_e$ depart from the true values r_B and $\dot{M}_B$, as a function of **r,** under the assumption that the Bondi solution **holds at all radii.**

The fiducial Bondi radius and mass accretion rate are then defined as:

$$r_e = \frac{G M_{BH}}{(c_s[r])^2}$$

$$\dot{M}_e = 4 \pi (r_e[r])^2 \lambda \rho[r] c_s[r] \quad (* \ 19 \ *)$$

In particular, r_e can be conveniently normalized to r_B as:

In[ ]:=
```
(*x=r/r_B;*)
(*Nc_s=c_s[r]/c_∞;*)
Nc_s[x] = Nρ^((γ-1)/2) (*7*)
(* == *)
r_B = (G M_BH)/(c_∞)^2; (*6*)
Nr_e = r_e / r_B
Nr_e2 = (Nc_s[x])^-2
(* Hacer algebra luego *)
Nr_e3 = ((x^2 𝓜)/λ)^((2 (γ-1))/(γ+1))  (*20*)
```

From (18)-(19):

In[ ]:=
```
Ṁ_B = 4 π r_B^2 λ ρ_∞ c_∞ (* 18 *)
(*Simplify[Ṁ_e/Ṁ_B]*)
(*21, manualmente*)
RatMeMB1 = (r_B/r_e)^((5-3 γ)/(2 (γ-1)))  (*21a*)
RatMeMB2 = FullSimplify[(1 / Nr_e3)^((5-3 γ)/(2 (γ-1)))] (*21b*)
RatMeMB = (r_B/re[x])^((5-3 γ)/(2 (γ-1)))  (*21*)
```

Obviously, for x → ∞, one has $r_e \to r_B$ (de 6 y 19):

In[ ]:=
```
r_e /. {c_s[r] → c_∞}
(*x→∞ se traduce en c_s[r]→c_∞ *)
r_B
```

Also seen by

In[ ]:=
```
Ṁ_e /. {r_e[r] → r_B, c_s[r] → c_∞, ρ[r] → ρ_∞}
(*Comparo!*)
Ṁ_B
```

From (10):

```
(*Λ=λ^(2(γ-1)/(γ+1))
   Eq22a=gGeneral[ℳ]==Λ fGeneral[x]  (*10*)
     (*== Ver *)
     fGeneral2=Normal[Series[ fGeneral[x],{x,0,1}]]
     gGeneral2=Series[gGeneral[ℳ],{ℳ,0,2}]*)
(*== De nuevo ... no me salio *)
Nr_e4 = Nr_e3 /. {ℳ → x^(-(5-3γ)/4)}
RatMeMB3 = RatMeMB2 /. {ℳ → x^(-(5-3γ)/4)}
```

Figure 1

```
(*Import[myDir<>"/Fig1a.png"]*)
```

```
Plot[Table[Nr_e4 /. {λ → 1, γ → i}, {i, 1.2, 1.2, 0.1}], {x, 10^-2, 10}]
```

Que raro no me da.. (debe ser que la Fig es hecha con un codigo de ellos!! ...)

```
λ_cr = (1 / 4) (2 / (5 - 3 γ)) ^((5-3 γ) / (2 (γ-1)));
(*14*)
RatReRB3 = (√2 / λ_cr)^(γ-1) x^((3 (γ-1))/2) ; (*22*)
RatReRB4 = (√2 / λ)^(γ-1) x^((3 (γ-1))/2)
```

```
test4 = Plot[Table[RatReRB3 /. {γ → i}, {i, 1.01, 1.6, 0.1}],
  {x, 10^-2, 15}, PlotRange → {{0.1, 6}, {0, 6}}]
```

```
test5 = LogPlot[
  Table[RatMeMB3 /. {λ → 1, γ → i}, {i, 1.2, 1.6, 0.1}], {x, 0.01, 10}]
```

For γ=5/3 there is an interesting result.

```
RatReRB4 /. {λ → 1 / 4, γ → 5 / 3}
```

For $\gamma = 5/3$ ($\lambda = 1/4$), there is an interesting result: $\boldsymbol{r_e} \sim 2^{5/3}\, r$, i.e., independently of the position r at which the temperature to derive $\boldsymbol{r_e}$ is taken, it is always concluded that the fiducial Bondi radius is placed at a larger radius (**r_e> r**), and by the same factor.

Adding the effects of electron scattering

```
Clear[L, l, χ]
```

$$L = \epsilon \dot{M}_{acc}\, c^2 \quad (*24*)$$

In the classical Bondi accretion $\dot{M}_{acc} = \dot{M}_B$.

For low accretion rates...

```
χ1 = 1 - l
l = L/LEdd (*25*)
```

La inclusion de la fuerza radiativa se puede hacer a traves de f(x) (11)

```
fRadi[x] = x^(4 (1-γ)/(γ+1)) (χ/x + 1/(γ - 1)) (*26*)
```

```
xRadi_min =
  Collect[Solve[D[fRadi[x], {x, 1}] == 0, x],
   χ][[1]] (*27*)
```

Being χ<1, este minimo esta mas cerca del BH que en el caso del Bondi sin radiation.

```
fRadi_min = fRadi[x] /. xRadi_min
```

```
test6 = LogLogPlot[{fGeneral[x] /. {γ → 1.4},
    fRadi[x] /. {γ → 1.4, χ → (1 - 0.7)}}, {x, 0, 10}]
```

Veo que la luminosidad L tiene que ser muy grande para que hayan diferencias entre f sin radiacion y f con Radiacion.

Since the minimum of g($\mathcal{M}$) is independent of electron scattering, the critical value of the new accretion parameter λ_{es}, at a given γ is:

```
λes = χ^2 λ (*28*)
```

Being $\chi < 1$, λ_{es} is lower than in the classical model. **The true accretion rate**, that we now call $\dot{M}_{es}$, is also reduced with respect to the classical value $\dot{M}_B$, for given M_{BH}, γ, and boundary conditions at infinity:

```
(*Ṁes=4π rB^2 λes ρ∞ c∞*) (* 29 *)
Ṁes = χ^2 ṀB (* 29 *)
```

From eqs. (29)-(30), one obtains explicitly $\dot{M}_{es} / \dot{M}_{Edd}$ in terms of $\dot{M}_B / \dot{M}_{Edd}$, by solving the quadratic equation:

```
(*Ṁes=(1-Ṁes/ṀEdd)^2 ṀB*) (* 31 *)
Eq31 = Mdotes == (1 - Mdotes / MdotEdd)^2 MdotB
```

Para algun momento del futuro

In[ ]:=
```
Solve[Eq31, Mdotes]
```

Tratando de obtener (32)

In[ ]:=
```
MdotesLow = Series[Solve[Eq31, Mdotes]〚1〛〚1〛〚2〛, {MdotB, 0, 4}]
```

Bueno es claro que para MdorB ~ 0, Mdotess ~ MdotB.

$$\dot{M}_{es} \Big/ \dot{M}_{Edd} \sim 1 - \sqrt{\left(\dot{M}_{Edd} \Big/ \dot{M}_{B}\right)} \quad (32)$$

We now apply to the Bondi solution with electron scattering the same procedure of Sect. 2.1, to quantify the differences, as a function of radius, between the true (r_B) and estimated (r_e) Bondi radius, and the true ($\dot{M}_{es}$) and estimated ($\dot{M}_e$) accretion rate, where r_e and $\dot{M}_e$ are defined as in eq. (19). It is easy to show that:

In[ ]:=
```
λes = χ^2 (1/4) (2/(5 - 3 γ))^((5-3 γ)/(2 (γ-1)));
(*28*)
RatResRB3 = (x^2 ℳ/λes)^((2 γ-1)/(γ+1)); (*33*)
RatResRB4 = RatResRB3 /. {ℳ → x^(-(5-3 γ)/4)}
RatMesMB3 = 1/χ^2 (1/RatResRB3)^((5-3 γ)/(2 (γ-1)))
RatMesMB4 =
  1/χ^2 (1/RatResRB4)^((5-3 γ)/(2 (γ-1))) (*34*)
```

Con y sin Radiacion

In[ ]:=
```
test7 = Plot[{
    RatReRB3 /. {γ → 1.4},
    RatResRB4 /. {γ → 1.4, χ → (1 - 0.1)}},
   {x, 10^-2, 15}, PlotRange → {{0.1, 6}, {0, 6}},
   PlotStyle → Thick,
   Frame → True,
   LabelStyle → Directive[Black, 18],
   FrameLabel → {Style["x=r/r_B", 24], Style["Ṁ_e/Ṁ_(B/es)", 24],
     Style["Fig 1: Comparison Ṁ con y sin radiacion",
      20]}, ImageSize → Large]
```

Tasas de acrecion con y sin radiacion

```
In[ ]:= Plot[{
  RatMeMB3 /. {λ → 1, γ → 1.4},
  RatMesMB4 /. {γ → 1.4, χ → (1 - 0.1)}},
 {x, 10^-2, 15}, PlotRange → {{0.1, 6}, {0, 6}}]
```

Bueno lo que en verdad necesito es el ARGUMENTO!!...

Para el 1er Caso de Paper, las a,b,c,d estan dadas allí. Y es para el **Isotermico:** $\lambda = e^{3/2} / 4$

```
In[ ]:= λSolIso = Limit[λcr, γ → 1]
```

```
In[ ]:= myArg1 = (a b (e^(-c/d + Y/d))^b / d) /.
  {a → 1/2, b → 2, c → 0, d → -1, Y → (1/x + 2 Log[x]) - Log[λSolIso]}
```

```
In[ ]:= DumpSave[myDir <> "/Paper4.mx", "Global`"]
```

Others functions

```
(*Remove["Global`*"]*)
(* Definition of my hamiltonian == *)
hamiltonian[V_]@psi_ := -ħ^2 / (2 m) D[psi, {x, 2}] + V psi
schroedingerD[V_]@psi_ := hamiltonian[V]@psi - (Energy * psi)
(* ======================================================*)
(* ======================================================*)
(* == Para mostrar los pasos == *)
ShowSteps[exp_] := WolframAlpha[ToString@HoldForm@InputForm@exp,
  {{"Input", 2}, "Content"}, PodStates → {"Input__Step-by-step solution"}]
SetAttributes[ShowSteps, HoldAll]
(*== Example *)
(*D[3*x[t]^3+6,t]//ShowSteps;*)
(* ======================================================*)
(* ======================================================*)
slogan[text1_, text2_] := Block[{s1, s2},
  maxwidth = Max[360, First@ImageDimensions@Rasterize@Text[Style[text1,
        FontSize → 30, Bold, Black, Background → None, FontFamily → "Broadway"]],
    First@ImageDimensions@Rasterize@Text[Style[text2, FontSize → 25,
        Bold, Black, Background → None, FontFamily → "Script MT Bold"]]];
  s1 = Graphics[{Text[Style[text1, FontSize → 30, Bold, Black,
        Background → None, FontFamily → "Broadway"]]}, ImageSize → maxwidth];
  s2 = Graphics[{Text[Style[text2, FontSize → 25, Bold, Black, Background → None,
        FontFamily → "Script MT Bold"]]}, ImageSize → maxwidth];
  ImageCompose[s1, s2, Scaled[{.5, 0.35}]]]
```

```
(* ====================================================*)
(* === Para escribir derivadas muy bonitas ==*)
pdConv[f_] :=
 TraditionalForm[f /. Derivative[inds__][g_][vars__] :→ Apply[Defer[D[g[vars], ##]] &,
     Transpose[{{vars}, {inds}}] /. {{var_, 0} :→ Sequence[], {var_, 1} :→ {var}}]]
(* ====================================================*)
(* ====================================================*)
(* == *)
(* == Y para Escribir la TISE bien en texto !! Para V=0  == *)
WriteHv1[V_]@psi_ :=
 If[V ≠ 0, TraditionalForm[-ℏ^2/(2 m)] × pdConv[D[psi, {x, 2}]] + TraditionalForm[V psi],
  TraditionalForm[-ℏ^2/(2 m)] × pdConv[D[psi, {x, 2}]]]
(* ====================================================*)
(* ====================================================*)
(* == Y para Escribir la TISE con
  potencial diferente de cero bien en texto !! == *)
WriteH1v1[V_]@psi_ :=
 TraditionalForm[-ℏ^2/(2 m)] × pdConv[D[psi, {x, 2}]] + TraditionalForm[V psi]
(* ====================================================*)
(* ====================================================*)

(* == Para derivadas bonitas == *)
DifferentialOperator /: MakeBoxes[DifferentialOperator[x__], form_] :=
 With[{sub = RowBox@BoxForm`MakeInfixForm[{x}, ",", form]},
  InterpretationBox[SubscriptBox["∂", sub], DifferentialOperator[x]]]
(* ====================================================*)
(* ====================================================*)
(* == Escribir el Gradiente en Esferica sin evaluar == *)
WriteGradientSphv1 :=
 (* == La parte radial == *)
 TraditionalForm[r̂] × DifferentialOperator[r] +
  (*== La parte theta == *)
  TraditionalForm[θ̂/r] × DifferentialOperator[θ] +
  (*== La parte phi == *)
  TraditionalForm[ϕ̂/(r Sin[θ])][DifferentialOperator[ϕ]]
(* ====================================================*)
(* ====================================================*)
(* ====================================================*)
(* ====================================================*)
(* == Escribir el Laplaciano en Esferica sin evaluar == *)
WriteLaplaSphv1 :=
 (* == La parte radial == *)
```

```
 TraditionalForm[1/r^2] × DifferentialOperator[r][(r^2 DifferentialOperator[r])] +
  (*== La parte theta == *)
  TraditionalForm[1/(r^2 Sin[θ])] ×
   DifferentialOperator[θ][(Sin[θ] DifferentialOperator[θ])] +
  (*== La parte phi == *)
  TraditionalForm[1/(r^2 Sin^2[θ])][(DifferentialOperator[ϕ])^2]
(* ====================================================*)
(* ====================================================*)
(* Definition of my Radial hamiltonian == *)
hamiltonianR[V_]@psi_ := D[(r^2 D[psi, {r, 1}]), r] - (2 m)/ℏ^2 (V - Ener) psi
schroedingerDR[V_]@psi_ := hamiltonianR[V]@psi - (l (l + 1) * psi)
(* ====================================================*)
(* ====================================================*)

(* ====================================================*)
(* ====================================================*)
(* Definition of my Radial hamiltonian, Pero para escribir == *)
hamiltonianRW[V_]@psi_ :=
 HoldForm[DifferentialOperator[r] (r^2 DifferentialOperator[r] psi)] -
  HoldForm[TraditionalForm[(2 m)/ℏ^2] × HoldForm[(V - Ener) psi]]
schroedingerDRW[V_]@psi_ := hamiltonianRW[V]@psi - (l (l + 1) * psi)
(* ====================================================*)
(* ====================================================*)
(* == Una buena funcion para subtituir variables y que opere adecuadamente *)
(* Estar pendiente, los detalles *)
F[func_, pot1_] := Block[{R, e}, e = schroedingerDR[pot1]@R[r] == 0;
    e /. {R → Function[r, #]}] &[func, pot1]
```

```
In[ ]:= hamiltonianR[V[r]]@(R[r] / r) /. {R[r] → u[r] / r, R'[r] → (r u'[r] - u[r]) / r^2}
```

```
In[ ]:= schroedingerDR[V[r]]@R[r] == 0
```

```
In[ ]:= hamiltonianRW[V[r]]@R[r] == l (l + 1) R[r]
        schroedingerDRW[V[r]]@R[r] == 0
```

```
In[ ]:= F[func_, pot1_] := Block[{R, e}, e = schroedingerDR[pot1]@R[r] == 0;
            e /. {R → Function[r, #]}] &[func, pot1]
```

```
In[ ]:= Eq1 =
         Collect[Collect[F[u[r] / r, (elec^2 / (4 π ϵ0 r))] // FullSimplify // Expand, r], u[r]] /.
          {Ener → (ℏ^2 / (2 m)) * k^2}
```

Para usar llaves grandes:
1. Ctrl 9 (inline cell);
2. esc pw esc
3. copy paste inside the pw element the output of something like Grid[{{a}, {b}, {c}}, Alignment -> Left]
4. substitute the content a, b, c with your content
5. Select the Grid [] ... and Crt+Boton Izquierdo ⟶ Evaluate in Place

```
DifferentialOperator /: MakeBoxes[DifferentialOperator[x__], form_] :=
 With[{sub = RowBox@BoxForm`MakeInfixForm[{x}, ",", form]},
  InterpretationBox[SubscriptBox["∂", sub], DifferentialOperator[x]]]
```