Documentation for KFK-INR SESAME Library

G.A. Moses and R.R. Peterson

September 1986

FPA-86-2
DOCUMENTATION FOR KfK-INR SESAME LIBRARY

G.A. Moses
R.R. Peterson

Fusion Power Associates
6515 Grand Teton Plaza
Madison, WI  53719

September 1986
1. Introduction

This documentation explains the implementation of the SESAME library provided to KfK-INR by Fusion Power Associates. It does not attempt to replace the SESAME documentation provided by LANL. In fact, the LANL documentation is included in the Appendix of this report.

The SESAME library from LANL consists of physical data in machine readable form and a set of FORTRAN subroutines that process the data and access the data tables. Table 1.1 gives the list of materials included in the GTDA library and the sub-libraries available for each material. The equation of state sub-library contains the pressure and specific internal energy information. The astrophysical sub-library contains the charge state and opacity data and the conductivity sub-library contains the electrical and thermal conductivity data. Generally, larger numbers in each sub-library indicate more recent data. These material data codes are referred to as SESAME material ID numbers. Material ID numbers in the equation of state library are four digit material ID numbers in the astrophysical library and five digits, starting with a 1, and material ID numbers in the conductivity library are five digits starting with a 2.

The GTDA library is unrestricted and does not require an export license. The complete SESAME library includes several deuterides and plastic explosives and requires an export license. Within each sub-library several different categories of data are available for each material and each data category is coded by the number of the table that holds it. These data category tables are described in Table 1.2.

Processing of the SESAME library is done in three steps. These are shown schematically in Fig. 1.1. The data is originally on a magnetic tape in ASCII
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>126</td>
<td>Actinium</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>741</td>
<td>Alumina</td>
<td>7410</td>
<td>17410</td>
<td>21261</td>
</tr>
<tr>
<td>371</td>
<td>Aluminum</td>
<td>3715</td>
<td></td>
<td></td>
</tr>
<tr>
<td>371</td>
<td>Aluminum</td>
<td>3711</td>
<td>13710</td>
<td>23713</td>
</tr>
<tr>
<td>371</td>
<td>Aluminum</td>
<td>3716</td>
<td></td>
<td></td>
</tr>
<tr>
<td>371</td>
<td>Aluminum</td>
<td>3712</td>
<td></td>
<td></td>
</tr>
<tr>
<td>371</td>
<td>Aluminum</td>
<td>3713</td>
<td></td>
<td></td>
</tr>
<tr>
<td>190</td>
<td>Americium</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>552</td>
<td>Ammonia</td>
<td>5520</td>
<td></td>
<td>21901</td>
</tr>
<tr>
<td>288</td>
<td>Antimony</td>
<td></td>
<td></td>
<td>22881</td>
</tr>
<tr>
<td>517</td>
<td>Argon</td>
<td>5171</td>
<td>15170</td>
<td>25171</td>
</tr>
<tr>
<td>517</td>
<td>Argon</td>
<td>5172</td>
<td></td>
<td></td>
</tr>
<tr>
<td>227</td>
<td>Arsenic</td>
<td></td>
<td></td>
<td>22271</td>
</tr>
<tr>
<td>144</td>
<td>Astadine</td>
<td></td>
<td></td>
<td>21441</td>
</tr>
<tr>
<td>208</td>
<td>Barium</td>
<td></td>
<td></td>
<td>22081</td>
</tr>
<tr>
<td>202</td>
<td>Beryllium</td>
<td>2022</td>
<td>12021</td>
<td>22021</td>
</tr>
<tr>
<td>202</td>
<td>Beryllium</td>
<td>2020</td>
<td>12020</td>
<td>22021</td>
</tr>
<tr>
<td>602</td>
<td>Beryllium Oxide</td>
<td></td>
<td>16020</td>
<td></td>
</tr>
<tr>
<td>322</td>
<td>Bismuth</td>
<td></td>
<td></td>
<td>23221</td>
</tr>
<tr>
<td>233</td>
<td>Boron</td>
<td></td>
<td></td>
<td>22331</td>
</tr>
<tr>
<td>708</td>
<td>Boron Carbide</td>
<td>7081</td>
<td></td>
<td></td>
</tr>
<tr>
<td>410</td>
<td>Brass</td>
<td>4100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>505</td>
<td>Bromine</td>
<td></td>
<td>15050</td>
<td>25051</td>
</tr>
<tr>
<td>553</td>
<td>Butane (normal)</td>
<td>5530</td>
<td></td>
<td></td>
</tr>
<tr>
<td>268</td>
<td>Cadmium</td>
<td></td>
<td></td>
<td>22681</td>
</tr>
<tr>
<td>203</td>
<td>Calcium</td>
<td></td>
<td>12030</td>
<td>22031</td>
</tr>
<tr>
<td>733</td>
<td>Calcium Carbonate</td>
<td>7330</td>
<td></td>
<td></td>
</tr>
<tr>
<td>783</td>
<td>Carbon Liquid</td>
<td>7831</td>
<td></td>
<td></td>
</tr>
<tr>
<td>754</td>
<td>Carbon Phenolic</td>
<td>7541</td>
<td>17540</td>
<td></td>
</tr>
<tr>
<td>720</td>
<td>Carbon Tetrafluoride</td>
<td></td>
<td>17200</td>
<td></td>
</tr>
<tr>
<td>317</td>
<td>Cerium</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>351</td>
<td>Cesium</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>502</td>
<td>Chlorine</td>
<td></td>
<td>15020</td>
<td>25021</td>
</tr>
<tr>
<td>585</td>
<td>Chlorine Oxide</td>
<td></td>
<td>15850</td>
<td></td>
</tr>
<tr>
<td>307</td>
<td>Chromium</td>
<td></td>
<td>13070</td>
<td>23071</td>
</tr>
<tr>
<td>312</td>
<td>Cobalt</td>
<td></td>
<td>13120</td>
<td>23121</td>
</tr>
<tr>
<td>333</td>
<td>Copper</td>
<td>3334</td>
<td></td>
<td></td>
</tr>
<tr>
<td>333</td>
<td>Copper</td>
<td>3333</td>
<td></td>
<td></td>
</tr>
<tr>
<td>333</td>
<td>Copper</td>
<td>3332</td>
<td></td>
<td></td>
</tr>
<tr>
<td>333</td>
<td>Copper</td>
<td>3330</td>
<td>13330</td>
<td>23333</td>
</tr>
<tr>
<td>333</td>
<td>Copper</td>
<td>3331</td>
<td></td>
<td></td>
</tr>
<tr>
<td>182</td>
<td>Curium</td>
<td></td>
<td></td>
<td>21821</td>
</tr>
<tr>
<td>799</td>
<td>DAP (Diallyl Phthalate)</td>
<td></td>
<td></td>
<td>17995</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>526</td>
<td>Deuterium</td>
<td>5263</td>
<td>15263</td>
<td></td>
</tr>
<tr>
<td>527</td>
<td>Deuterium-Tritium</td>
<td>5271</td>
<td></td>
<td></td>
</tr>
<tr>
<td>783</td>
<td>Diamond</td>
<td>7830</td>
<td>17830</td>
<td>27831</td>
</tr>
<tr>
<td>503</td>
<td>Dry Air</td>
<td>5030</td>
<td>15030</td>
<td></td>
</tr>
<tr>
<td>212</td>
<td>Dysprosium</td>
<td></td>
<td>12120</td>
<td>22121</td>
</tr>
<tr>
<td>760</td>
<td>Epoxy (Epon 828)</td>
<td>7602</td>
<td></td>
<td></td>
</tr>
<tr>
<td>760</td>
<td>Epoxy (Epon 828)</td>
<td>7601</td>
<td>17600</td>
<td></td>
</tr>
<tr>
<td>285</td>
<td>Erbium</td>
<td></td>
<td></td>
<td>22851</td>
</tr>
<tr>
<td>211</td>
<td>Europium</td>
<td>2110</td>
<td>12110</td>
<td>22111</td>
</tr>
<tr>
<td>765</td>
<td>Fiberglass</td>
<td></td>
<td>17650</td>
<td></td>
</tr>
<tr>
<td>504</td>
<td>Fluorine</td>
<td></td>
<td>15040</td>
<td>25041</td>
</tr>
<tr>
<td>691</td>
<td>Formaldehyde</td>
<td></td>
<td>16910</td>
<td></td>
</tr>
<tr>
<td>134</td>
<td>Francium</td>
<td></td>
<td></td>
<td>21341</td>
</tr>
<tr>
<td>328</td>
<td>Gadolinium</td>
<td>3280</td>
<td></td>
<td>23281</td>
</tr>
<tr>
<td>237</td>
<td>Gallium</td>
<td></td>
<td></td>
<td>22371</td>
</tr>
<tr>
<td>395</td>
<td>Germanium</td>
<td></td>
<td></td>
<td>23951</td>
</tr>
<tr>
<td>270</td>
<td>Gold</td>
<td>2700</td>
<td></td>
<td>22701</td>
</tr>
<tr>
<td>389</td>
<td>Hafnium</td>
<td></td>
<td></td>
<td>23891</td>
</tr>
<tr>
<td>576</td>
<td>Helium</td>
<td>5762</td>
<td></td>
<td></td>
</tr>
<tr>
<td>576</td>
<td>Helium</td>
<td>5760</td>
<td>15760</td>
<td>25761</td>
</tr>
<tr>
<td>576</td>
<td>Helium</td>
<td>5761</td>
<td></td>
<td></td>
</tr>
<tr>
<td>821</td>
<td>HE-9404</td>
<td></td>
<td>18210</td>
<td></td>
</tr>
<tr>
<td>366</td>
<td>Holmium</td>
<td>3660</td>
<td></td>
<td>23661</td>
</tr>
<tr>
<td>525</td>
<td>Hydrogen</td>
<td>5251</td>
<td></td>
<td></td>
</tr>
<tr>
<td>525</td>
<td>Hydrogen</td>
<td>5250</td>
<td>15250</td>
<td>25251</td>
</tr>
<tr>
<td>321</td>
<td>Indium</td>
<td></td>
<td></td>
<td>23211</td>
</tr>
<tr>
<td>507</td>
<td>Iodine</td>
<td></td>
<td>15070</td>
<td>25071</td>
</tr>
<tr>
<td>374</td>
<td>Iridium</td>
<td></td>
<td></td>
<td>23741</td>
</tr>
<tr>
<td>214</td>
<td>Iron</td>
<td>2145</td>
<td></td>
<td></td>
</tr>
<tr>
<td>214</td>
<td>Iron</td>
<td>2140</td>
<td>12140</td>
<td>22143</td>
</tr>
<tr>
<td>554</td>
<td>Isobutane</td>
<td>5540</td>
<td></td>
<td></td>
</tr>
<tr>
<td>518</td>
<td>Krypton</td>
<td>5181</td>
<td></td>
<td></td>
</tr>
<tr>
<td>518</td>
<td>Krypton</td>
<td>5180</td>
<td>15180</td>
<td>25181</td>
</tr>
<tr>
<td>388</td>
<td>Lanthanum</td>
<td></td>
<td></td>
<td>23881</td>
</tr>
<tr>
<td>320</td>
<td>Lead</td>
<td>3200</td>
<td></td>
<td>23201</td>
</tr>
<tr>
<td>725</td>
<td>Lithia-Boria Glass</td>
<td>7252</td>
<td>17250</td>
<td></td>
</tr>
<tr>
<td>229</td>
<td>Lithium</td>
<td>2290</td>
<td>12290</td>
<td>22291</td>
</tr>
<tr>
<td>229</td>
<td>Lithium</td>
<td>2291</td>
<td></td>
<td></td>
</tr>
<tr>
<td>724</td>
<td>Lithium Deutride</td>
<td>7243</td>
<td>17241</td>
<td></td>
</tr>
<tr>
<td>724</td>
<td>Lithium Deuteride</td>
<td>7241</td>
<td>17240</td>
<td></td>
</tr>
<tr>
<td>735</td>
<td>Lithium Fluoride</td>
<td></td>
<td>17350</td>
<td></td>
</tr>
<tr>
<td>737</td>
<td>Lithium Hydride</td>
<td>7371</td>
<td>17370</td>
<td></td>
</tr>
<tr>
<td>289</td>
<td>Lutetium</td>
<td></td>
<td></td>
<td>22891</td>
</tr>
<tr>
<td>286</td>
<td>Magnesium</td>
<td></td>
<td>12860</td>
<td>22861</td>
</tr>
<tr>
<td>308</td>
<td>Manganese</td>
<td></td>
<td>13080</td>
<td>23081</td>
</tr>
<tr>
<td>332</td>
<td>Mercury</td>
<td></td>
<td></td>
<td>23321</td>
</tr>
<tr>
<td>550</td>
<td>Methane</td>
<td>5502</td>
<td></td>
<td></td>
</tr>
<tr>
<td>550</td>
<td>Methane</td>
<td>5500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>550</td>
<td>Methane</td>
<td>5501</td>
<td></td>
<td></td>
</tr>
<tr>
<td>752</td>
<td>Mica</td>
<td>7520</td>
<td></td>
<td></td>
</tr>
<tr>
<td>298</td>
<td>Molybdenum</td>
<td>2982</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number</td>
<td>Substance</td>
<td>Year 1</td>
<td>Year 2</td>
<td>Year 3</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>298</td>
<td>Molybdenum</td>
<td>2981</td>
<td>12980</td>
<td>22981</td>
</tr>
<tr>
<td>298</td>
<td>Molybdenum</td>
<td>2980</td>
<td>12980</td>
<td></td>
</tr>
<tr>
<td>298</td>
<td>Molybdenum</td>
<td>2983</td>
<td></td>
<td></td>
</tr>
<tr>
<td>755</td>
<td>Mylar</td>
<td></td>
<td>17550</td>
<td></td>
</tr>
<tr>
<td>311</td>
<td>Neodymium</td>
<td></td>
<td></td>
<td>23111</td>
</tr>
<tr>
<td>541</td>
<td>Neon</td>
<td>5411</td>
<td></td>
<td></td>
</tr>
<tr>
<td>541</td>
<td>Neon</td>
<td>5410</td>
<td>15410</td>
<td>25411</td>
</tr>
<tr>
<td>164</td>
<td>Neptunium</td>
<td></td>
<td></td>
<td>21641</td>
</tr>
<tr>
<td>711</td>
<td>Nevada Alluvium</td>
<td>7111</td>
<td>17111</td>
<td></td>
</tr>
<tr>
<td>310</td>
<td>Nickel</td>
<td>3100</td>
<td>13100</td>
<td>23101</td>
</tr>
<tr>
<td>274</td>
<td>Niobium</td>
<td></td>
<td>12740</td>
<td>22741</td>
</tr>
<tr>
<td>500</td>
<td>Nitrogen</td>
<td>5000</td>
<td>15000</td>
<td>25001</td>
</tr>
<tr>
<td>500</td>
<td>Nitrogen</td>
<td>5001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>Osmium</td>
<td></td>
<td></td>
<td>22561</td>
</tr>
<tr>
<td>501</td>
<td>Oxygen</td>
<td>5011</td>
<td></td>
<td></td>
</tr>
<tr>
<td>501</td>
<td>Oxygen</td>
<td>5010</td>
<td>15010</td>
<td>25011</td>
</tr>
<tr>
<td>383</td>
<td>Palladium</td>
<td></td>
<td></td>
<td>23831</td>
</tr>
<tr>
<td>762</td>
<td>Parylene (Polymonochloroparaxylylene)</td>
<td>17620</td>
<td></td>
<td></td>
</tr>
<tr>
<td>822</td>
<td>PBX-9501 (High Explosive)</td>
<td></td>
<td>18220</td>
<td></td>
</tr>
<tr>
<td>820</td>
<td>PBX-9502 (High Explosive)</td>
<td>8200</td>
<td>18201</td>
<td></td>
</tr>
<tr>
<td>722</td>
<td>Phosphorous Trifluoride</td>
<td></td>
<td>17220</td>
<td></td>
</tr>
<tr>
<td>391</td>
<td>Phosphorus</td>
<td></td>
<td>13910</td>
<td>23911</td>
</tr>
<tr>
<td>373</td>
<td>Platinum</td>
<td>3730</td>
<td></td>
<td>23731</td>
</tr>
<tr>
<td>119</td>
<td>Plutonium</td>
<td></td>
<td></td>
<td>21191</td>
</tr>
<tr>
<td>191</td>
<td>Polonium</td>
<td></td>
<td></td>
<td>21911</td>
</tr>
<tr>
<td>738</td>
<td>Polycrystal Quartz</td>
<td>7383</td>
<td></td>
<td></td>
</tr>
<tr>
<td>717</td>
<td>Polyethylene (branched)</td>
<td>7171</td>
<td>17170</td>
<td></td>
</tr>
<tr>
<td>716</td>
<td>Polyethylene (branched, fully deuterated)</td>
<td>7160</td>
<td></td>
<td></td>
</tr>
<tr>
<td>718</td>
<td>Polyethylene (Marlex)</td>
<td>7180</td>
<td></td>
<td></td>
</tr>
<tr>
<td>759</td>
<td>Polystyrene</td>
<td>7590</td>
<td>17590</td>
<td></td>
</tr>
<tr>
<td>759</td>
<td>Polystyrene Foam</td>
<td></td>
<td>17591</td>
<td></td>
</tr>
<tr>
<td>759</td>
<td>Polystyrene + Krypton</td>
<td></td>
<td>17592</td>
<td></td>
</tr>
<tr>
<td>723</td>
<td>Polytetradeuteropentethylene (linear)</td>
<td>7230</td>
<td></td>
<td></td>
</tr>
<tr>
<td>756</td>
<td>Polyurethane</td>
<td>7560</td>
<td>17560</td>
<td></td>
</tr>
<tr>
<td>246</td>
<td>Potassium</td>
<td></td>
<td>12460</td>
<td>22461</td>
</tr>
<tr>
<td>210</td>
<td>Praseodymium</td>
<td></td>
<td>12100</td>
<td></td>
</tr>
<tr>
<td>295</td>
<td>Promethium</td>
<td></td>
<td></td>
<td>22951</td>
</tr>
<tr>
<td>138</td>
<td>Protactinium</td>
<td></td>
<td></td>
<td>21381</td>
</tr>
<tr>
<td>738</td>
<td>Quartz</td>
<td>7381</td>
<td></td>
<td></td>
</tr>
<tr>
<td>738</td>
<td>Quartz</td>
<td>7380</td>
<td>17380</td>
<td></td>
</tr>
<tr>
<td>158</td>
<td>Radium</td>
<td></td>
<td></td>
<td>21581</td>
</tr>
<tr>
<td>506</td>
<td>Radon</td>
<td></td>
<td></td>
<td>25061</td>
</tr>
<tr>
<td>304</td>
<td>Rhenium</td>
<td></td>
<td></td>
<td>23041</td>
</tr>
<tr>
<td>305</td>
<td>Rhodium</td>
<td></td>
<td></td>
<td>23051</td>
</tr>
<tr>
<td>281</td>
<td>Rubidium</td>
<td></td>
<td></td>
<td>22811</td>
</tr>
<tr>
<td>275</td>
<td>Ruthenium</td>
<td></td>
<td></td>
<td>22751</td>
</tr>
<tr>
<td>728</td>
<td>Salt (Sodium Chloride)</td>
<td>7281</td>
<td>17280</td>
<td></td>
</tr>
<tr>
<td>394</td>
<td>Samarium</td>
<td></td>
<td></td>
<td>23941</td>
</tr>
<tr>
<td>301</td>
<td>Selenium</td>
<td></td>
<td></td>
<td>23011</td>
</tr>
<tr>
<td>381</td>
<td>Silicon</td>
<td></td>
<td></td>
<td>23811</td>
</tr>
<tr>
<td>Substance</td>
<td>Address</td>
<td>Contact</td>
<td>Code</td>
<td></td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>---------</td>
<td>---------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>Silicon Dioxide</td>
<td>738</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Silver</td>
<td>272</td>
<td></td>
<td>12720</td>
<td></td>
</tr>
<tr>
<td>Sodium</td>
<td>244</td>
<td></td>
<td>12448</td>
<td></td>
</tr>
<tr>
<td>Solar Mixture (Ross-Aller)</td>
<td>528</td>
<td></td>
<td>15280</td>
<td></td>
</tr>
<tr>
<td>Spinel</td>
<td>740</td>
<td></td>
<td>17400</td>
<td></td>
</tr>
<tr>
<td>Stainless Steel (304)</td>
<td>427</td>
<td></td>
<td>14270</td>
<td></td>
</tr>
<tr>
<td>Strontium</td>
<td>360</td>
<td></td>
<td>23601</td>
<td></td>
</tr>
<tr>
<td>Sulfur Dioxide</td>
<td>581</td>
<td></td>
<td>15810</td>
<td></td>
</tr>
<tr>
<td>Sulfur Fluoride</td>
<td>583</td>
<td></td>
<td>15830</td>
<td></td>
</tr>
<tr>
<td>Sulphur</td>
<td>401</td>
<td></td>
<td>14010</td>
<td></td>
</tr>
<tr>
<td>Sylgard (Silicon Rubber)</td>
<td>793</td>
<td></td>
<td>17930</td>
<td></td>
</tr>
<tr>
<td>Tantalum</td>
<td>352</td>
<td></td>
<td>23521</td>
<td></td>
</tr>
<tr>
<td>Technetium</td>
<td>102</td>
<td></td>
<td>21021</td>
<td></td>
</tr>
<tr>
<td>Teflon (Polytetrafluoroethylene)</td>
<td>719</td>
<td></td>
<td>17190</td>
<td></td>
</tr>
<tr>
<td>Tellurium</td>
<td>297</td>
<td></td>
<td>22971</td>
<td></td>
</tr>
<tr>
<td>Terbium</td>
<td>222</td>
<td></td>
<td>22221</td>
<td></td>
</tr>
<tr>
<td>Thallium</td>
<td>370</td>
<td></td>
<td>23701</td>
<td></td>
</tr>
<tr>
<td>Thorium</td>
<td>141</td>
<td></td>
<td>21411</td>
<td></td>
</tr>
<tr>
<td>Thulium</td>
<td>338</td>
<td></td>
<td>23381</td>
<td></td>
</tr>
<tr>
<td>Tin</td>
<td>216</td>
<td></td>
<td>12160</td>
<td></td>
</tr>
<tr>
<td>Titanium</td>
<td>296</td>
<td></td>
<td>12960</td>
<td></td>
</tr>
<tr>
<td>Tuff</td>
<td>712</td>
<td></td>
<td>17120</td>
<td></td>
</tr>
<tr>
<td>Tungsten</td>
<td>354</td>
<td></td>
<td>3541</td>
<td></td>
</tr>
<tr>
<td>Tungsten Carbide</td>
<td>356</td>
<td></td>
<td>3560</td>
<td></td>
</tr>
<tr>
<td>Uranium</td>
<td>154</td>
<td></td>
<td>1540</td>
<td></td>
</tr>
<tr>
<td>Uranium Dioxide</td>
<td>743</td>
<td></td>
<td>7432</td>
<td></td>
</tr>
<tr>
<td>Urethane</td>
<td>753</td>
<td></td>
<td>17530</td>
<td></td>
</tr>
<tr>
<td>Vanadium</td>
<td>255</td>
<td></td>
<td>22551</td>
<td></td>
</tr>
<tr>
<td>Water</td>
<td>715</td>
<td></td>
<td>7150</td>
<td></td>
</tr>
<tr>
<td>Water</td>
<td>715</td>
<td></td>
<td>7152</td>
<td></td>
</tr>
<tr>
<td>Westerly Granite</td>
<td>739</td>
<td></td>
<td>7390</td>
<td></td>
</tr>
<tr>
<td>Xenon</td>
<td>519</td>
<td></td>
<td>5190</td>
<td></td>
</tr>
<tr>
<td>Ytterbium</td>
<td>344</td>
<td></td>
<td>23441</td>
<td></td>
</tr>
<tr>
<td>Yttrium</td>
<td>236</td>
<td></td>
<td>22361</td>
<td></td>
</tr>
<tr>
<td>Zinc</td>
<td>314</td>
<td></td>
<td>13140</td>
<td></td>
</tr>
<tr>
<td>Zirconium</td>
<td>318</td>
<td></td>
<td>23181</td>
<td></td>
</tr>
<tr>
<td>Category #</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>101-199</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>201</td>
<td>Basic data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>301</td>
<td>Total EOS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>303</td>
<td>Ion EOS plus cold curve</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>304</td>
<td>Electron EOS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>305</td>
<td>Ion EOS (including zero point)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>306</td>
<td>Cold curve (no zero point)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>401</td>
<td>Vaporization</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>501</td>
<td>Opacity grid boundary: calculated vs. interpolated</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>502</td>
<td>Rosseland mean opacity (cm² g⁻¹)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>503</td>
<td>Electron conductive opacity¹ (cm² g⁻¹)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>504</td>
<td>Mean ion charge¹ (free electrons per atom)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>505</td>
<td>Plank mean opacity (cm² g⁻¹)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>601</td>
<td>Mean ion charge² (free electrons per atom)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>602</td>
<td>Electrical conductivity (sec⁻¹)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>603</td>
<td>Thermal conductivity (cm⁻¹ sec⁻¹)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>604</td>
<td>Thermoelectric coefficient (cm⁻¹ sec⁻¹)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>605</td>
<td>Electron conductive opacity² (cm² g⁻¹)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

¹Opacity model (Hubbard-Lampe)
²Conductivity model (Ziman)
Fig. 1.1. Processing of SESAME Data
or EBCDIC format. This data is processed by the UPDATE program to create a SESAME library in the binary format of the host computer. This is step one. This step is normally performed only once, but can be used again to update the binary library with new information. Once the SESAME binary library is created, the ASCII or EBCDIC data is no longer used and all further processing is done on the binary library.

To use the SESAME library in an application program such as SIMMER or MEDUSA-KA, a set of "GET ..." subroutines must be included in the application program. This is the second processing step. These subroutines access the SESAME binary library and read in only the material properties requested by the user. This is done once at the beginning of a calculation.

The third step is interpolation in the tables. This is done by calling either SESAME or KfK interpolation routines and passing the two independent variables defining the interpolation point and identifying the data table to interpolate upon. This third step will occur every time that material property data is required, usually several times on each time step.

The remainder of this report is written to more fully describe each of these three steps, with the aim of allowing easy implementation of this SESAME library.
2. Creating and Updating the SESAME Binary Library from Character Files

2.1 Conversion of Character SESAME Files into a Binary Library

The data tables that make up the SESAME libraries come from Los Alamos National Laboratory in the form of several large character files. The libraries must be changed into a binary file that is in the proper format to be used by the routines discussed in Chapter 3. The format of the character files is summarized in Table 2.1; the format of the binary library is given in Fig. 3.4 and Tables 3.3 through 3.7. We will now discuss the character files and the tapes they come from in more detail and conversion of the data into binary files; the details of the binary files will be left to Chapter 3.

2.2 SESAME Tapes and Data Files

The character files are broken into 8 files, which are written on the provided tapes. The two tapes, externally labeled SES I and SES II, are written in ASCII, with a record size of 80 characters, with 100 records per block, and at a density of 1600 bpi. The first four files, EOS1, EOS2, EOS3, and EOS4, contain equation-of-state tables. They should be merged together to make one large file. The next two files, OP1 and OP2, contain the opacity data and should also be merged together. These six files are on the first tape. The second SESAME tape, SES II, contains the conductivity data in two files, COND1 and COND2, which must be merged. In fact, it is much easier to create the binary library file if all of the files are combined into one large character file. To do this, one must be sure to remove all of the blank lines and the lines with only a "2" in columns 2 and 80 at the end of EOS4 and OP2. The final line in any character files that are operated on by CREATELIB must contain a "2" in columns 2 and 80, so do not remove this line from the end of COND2, but do remove any blank lines.
Table 2.1. Format of SESAME Character Files

<table>
<thead>
<tr>
<th>Position of Lines</th>
<th>Items in Lines</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>first line for</td>
<td>.&quot;0&quot;</td>
<td>(I2,2F6.0,I6,3X, A1,2F9.0,F4.0)</td>
</tr>
<tr>
<td>each material</td>
<td>.material id</td>
<td></td>
</tr>
<tr>
<td></td>
<td>.&quot;101&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>.NWDC = # of words in table</td>
<td></td>
</tr>
<tr>
<td></td>
<td>.character if = &quot;D&quot;, then this line followed by a replacement line</td>
<td></td>
</tr>
<tr>
<td></td>
<td>.creation date</td>
<td></td>
</tr>
<tr>
<td></td>
<td>.date of last update</td>
<td></td>
</tr>
<tr>
<td></td>
<td>.version #</td>
<td></td>
</tr>
</tbody>
</table>

| next NWDC/80 lines | .NWDC characters describing the contents of the data set for this material | (20A4) |

| next line | ."1" | (I2,2F6.0,I6,3X, A1,2F9.0,F4.0) |
|           | .material id |        |
|           | ."102" |        |
|           | .NWDC = # of words in table |        |
|           | .character if = "D", then this line followed by a replacement line |        |
|           | .creation date |        |
|           | .date of last update |        |
|           | .version # |        |

| next NWDC/80 lines | .NWDC characters describing the contents of the data set for this material | (20A4) |

| next line | ."1" | (I2,2F6.0,I6,3X, A1,2F9.0,F4.0) |
|           | .material id |        |
|           | .table id |        |
|           | .NWDC = # of words in table |        |
|           | .character if = "D", then this line followed by a replacement line |        |
|           | .creation date |        |
|           | .date of last update |        |
|           | .version # |        |

| next NWDC/80 lines | .5 data words | (5E15.0,5I1) |
|                   | ."11111" |        |
|                   | .5 data words |        |
|                   | ."11111" |        |
|                   | |        |
|                   | |        |


.n data words (n<5)
.n "1"s, 5-n "0"s

............ repeat for additional materials ............

final line ."2" (I2)
There are two materials for which the data is not in a proper form that we suggest be deleted from the large character file before CREATELIB operates on it. These are two equation-of-state tables with id numbers 3715 (aluminum) and 3334 (copper). The loss of this data does not pose a major limitation to the library because we have been given 5 EOS sets for both aluminum and copper. If any other problems appear in the data, such as an improper character due to a dropped bit, we suggest that the user replace that character, if it is obvious what its value should be. If its value is unclear or if there are many such problems with a particular set of data, we suggest that that set also be removed before running CREATELIB. To remove a set of data from a file, one should be sure to remove a block starting at a line with a "0" in columns 2 and 80 and ending one line before the next line that is constructed in this manner.

Following the data files on tape SES II are three files containing programs. The first is a file called CREATELIB that contains the main program CREATELIB plus all of the subroutines from Los Alamos National Laboratory that it calls. Next is a file called GETDATA that contains many subroutines required to get data out of a binary library file and to perform tasks such as interpolation on the data that has been extracted from the library. Finally there is one file, SESPAK, that contains all the routines provided by Los Alamos National Laboratory with the SESAME data tables.

2.3 The CREATELIB Code to Create a Binary SESAME Library

The CREATELIB code has been written to create a sequential binary library file from a character file. The code uses a group of routines that are part of the SESAME subroutine package. A listing of the code is given in Appendix A. This code is written to run on a Digital Equipment Corp. VAX computer,
though very few changes are required to make it run on any computer. These changes may include modifications to the form of the OPEN statement, the file names, and possibly the ENDFILE statement. To use this code, one must assign the character file containing the data from the SESAME tapes to SESIN.DAT and the new binary file will appear as NEWLIB.DAT. Another file, SESTEMP.DAT, is used as a temporary scratch file. The file CURLIB.DAT, may have to be assigned but it is not used if one is creating a new library and not modifying an old one. Obviously, these filenames must be adapted to conform to the conventions of the machine being used.

2.4 Updating an Existing Binary SESAME Library

The CREATELIB routine may be used to add new data tables to an existing binary SESAME library. The new data files to be added are in the same form as the original character SESAME files, as discussed in Sections 2.1 and 2.2. One can modify the CREATELIB program to update an existing binary library by changing the parameter INEW from 0 to 1. The code is run in a manner similar to that described in Section 2.3, with the old binary library file being CURLIB.DAT, the new binary library file being NEWLIB.DAT, and the data added to the old library to make the new one being in the character file SESIN.DAT.
3. Inputting Selected Material Properties into an Application Program

Radiation-hydrodynamics computer codes for ICF applications generally require the electron and ion pressure and specific internal energy as functions of density and temperature for a range of density from a fraction of solid to as much as thousands of times solid density. The temperature can range from nearly 0°C to hundreds of kilovolts. Such codes also require the plasma charge state (number of free electrons per atom) and the Rosseland and Planck opacities as functions of density and temperature. An integrated set of subroutines has been developed to input the total plasma pressure and specific internal energy, the charge state, and the Rosseland/Planck opacities for user specified materials. Inclusion of these subroutines in the initialization part of MEDUSA-KA for instance will allow subsequent interpolation in the data tables during the hydrodynamic simulation.

Test reactor safety hydrodynamic simulation codes such as SIMMER require similar material property data. Only in this case, the temperature and density range is much more restricted. The SESAME library contains tables especially suited for reactor safety applications for relevant materials such as sodium, steel, and water. The subroutines provided with the KfK-INR SESAME library can be adapted to either of these two applications.

In the following subsections the input requirements to the subroutines and the details of the data formats are described. Listings of all subroutines used to retrieve the data and interpolate it are given in Appendix B.

3.1 Utilizing the GETLIB Subroutine

The package of subroutines that is used to retrieve the specified data is shown in Fig. 3.1. The interface routine to the hydro code is GETLIB. This was written by FPA and was not included in the routines from LANL. A single
Fig. 3.1. Structure Chart of GETLIB Subroutine Package
call to GETLIB will retrieve all data required for the subsequent calculation. The argument list for GETLIB is given in Table 3.1. The COMMON blocks required by these routines are given in Table 3.2. A further explanation of the MATS and UCONV variables is necessary.

MATS -- This is a matrix dimensioned MATS(2,NREG) that is used to specify the (1) equation of state SESAME material ID and the (2) astrophysical SESAME material ID for each of the materials used in a calculation. This is best understood by example. Suppose we want to simulate the ion beam target shown in Fig. 3.2. There are three materials in the problem. We will denote DT as material #1, Be as material #2, and Fe as material #3. To retrieve the data for these three materials we would specify

\[
\begin{align*}
\text{MATS}(1,1) &= 5271 \quad \text{MATS}(2,1) = 15263 \quad \text{ICAT}(1,1) \\
\text{MATS}(1,2) &= 2020 \quad \text{MATS}(2,2) = 12020 \quad \text{ICAT}(1,1) \\
\text{MATS}(1,3) &= 2140 \quad \text{MATS}(2,3) = 12140 \quad \text{ICAT}(1,1)
\end{align*}
\]

The first SESAME material ID will specify the pressure and energy data from the equation of state sub-library and the second ID will specify the charge state and opacity data from the astrophysical sub-library. Notice that there is no restriction on the choice of these ID's. Data from each of the sub-libraries could be defined to create a fictitious material. Either ID can be left as zero and no data will be retrieved. Both ID's equal to zero signifies the end of the list of materials.

UCONV(3,5) -- This a matrix of unit conversion factors to convert the table units from the SESAME values to those used by the host computer code. Some values of UCONV are redundant and must be equal to one another. The first index is for the two independent and one dependent variables (1 - density, 2 - temperature, 3 - dependent). The second index is for the
Table 3.1. Argument List for GETLIB Subroutine

LENA -- Length of the TBLS vector in double words (Input)

LCNT -- Pointer into TBLS vector (Input/Output)

NMATS -- Number of materials processed $1 \leq \text{NMATS} \leq 20$ (Output)

MATS(2,10) -- Matrix of equation of state and astrophysical material ID numbers for up to 10 materials (Input)

NWDES -- Total number of double words used in TBLS vector for the data tables (Output)

UCONV(3,5) -- unit conversion factors for the independent and dependent variables

  UCONV(1,1) -- Density variable for pressure
  UCONV(2,1) -- Temperature variable for pressure
  UCONV(3,1) -- Pressure variable
  UCONV(1,2) -- Density variable for internal energy (must = UCONV(1,1))
  UCONV(2,2) -- Temperature variable for internal energy (must = UCONV(2,1))
  UCONV(3,2) -- Internal energy variable
  UCONV(1,3) -- Density variable for charge state
  UCONV(2,3) -- Temperature variable for charge state
  UCONV(3,3) -- Charge state
  UCONV(1,4) -- Density variable for Rosseland opacity
  UCONV(2,4) -- Temperature variable for Rosseland opacity
  UCONV(3,4) -- Rosseland Opacity
  UCONV(1,5) -- Density variable for Planck opacity (must = UCONV(1,4))
  UCONV(2,5) -- Temperature variable for Planck opacity (must = UCONV(2,4))
  UCONV(3,5) -- Planck opacity

IRAD -- = 0 no opacity tables are loaded (Input)
    = 1 opacity tables are loaded
Table 3.1. (continued)

LES -- logical I/O unit number for equation of state sub-library (Input)

LOP -- logical I/O unit number for astrophysical sub-library (Input)

LOUT -- logical I/O unit number for printed output (Input)

LONGIO -- = 0 brief summary of loaded tables in written
= 1 full listing of table contents is written (Input)
Table 3.2. COMMON Blocks Used by GETLIB Package

COMMON/S2DIR/

LCMX -- Maximum number of double words available in TBLS for storing data tables

NRS -- Maximum number of materials that can be loaded

LCFW(NREG,NDAT) -- Pointers to data tables in TBLS for NREG materials and NDAT data types per material with (1) Total P&E, (2) Z, (3) Opacities, (4) Nuclear P&E, (5) Electron P&E.

Currently NREG=20 and NDAT=5.

COMMON/SESDAT/

TBLS(15000) -- REAL*8 vector where data tables are stored
Fig. 3.2 Ion Beam Target.
data category (1 - pressure, 2 - internal energy, 3 - charge state,
4 - Rosseland opacity, 5 - Planck opacity). The values must be inputted
such that

\[
\begin{align*}
\text{UCONV}(1,1) &= \text{UCONV}(1,2) \\
\text{UCONV}(2,1) &= \text{UCONV}(2,2) \\
\text{UCONV}(1,4) &= \text{UCONV}(1,5) \\
\text{UCONV}(2,4) &= \text{UCONV}(2,5)
\end{align*}
\]

Figure 3.3 shows an example test program that calls GETLIB. The output
from GETLIB is also shown. This test program retrieves the data for beryllium
and outputs the data using the LONGIO=1 option.

3.1.2 Data Formats in the GET... Process

The format of the data is different in the SESAME binary library than it
is in the data tables used by the application program. The GET... routines
retrieve the specified data and convert it into the format for use in the
code.

The format of the data in the SESAME library is shown in Fig. 3.4 and is
discussed in LANL documentation (Appendix C). Its details will not be dis-
cussed here. However, it is important to note that the data in the SESAME bi-
nary library is written in single precision format with each 32-bit word con-
taining one number.

The units of the data in the SESAME binary library are different for each
different sublibrary. These units are given in Table 3.3.

3.1.2.1 GETEOS -- Equation of State Data

Basic material data is stored in the 201. table for each material in the
SESAME binary library. The format of this data is given in Table 3.4. The
Fig. 3.3. Test Program that calls GETLIB and EOSTAB to Retrieve and Interpolate in Data Tables

C TEST PROGRAM FOR SESAME LIBRARY

C VARIABLES:
C T TEMPERATURE VECTOR
C D DENSITY VECTOR
C P PRESSURE VECTOR
C E SPECIFIC INTERNAL ENERGY VECTOR
C DPDR DENSITY DERIVATIVE OF PRESSURE VECTOR
C DPDT TEMPERATURE DERIVATIVE OF PRESSURE VECTOR
C DEDR DENSITY DERIVATIVE OF SPECIFIC INTERNAL ENERGY
C DEDT TEMPERATURE DERIVATIVE OF SPECIFIC INTERNAL ENERGY
C Z CHARGE STATE VECTOR
C SP PLANCK OPACITY VECTOR
C SR ROSSELAND OPACITY VECTOR
C UCONV UNIT CONVERSION FACTOR MATRIX
C IMAT MATERIAL INDEX VECTOR
C MATS SESAME MATERIAL ID NUMBER INPUT MATRIX

DIMENSION MATS(2,20)
DIMENSION IMAT(20), T(20), D(20), P(20), DPDR(20), DPDT(20),
1 E(20), DEDR(20), DEDT(20), Z(20), SR(20),
2 SP(20), XLOG(20), YLOG(20), UCONV(3,5)

C DEFINE THE VECTOR TO HOLD THE SESAME LIBRARY
REAL*8 A
COMMON/SESDAT/ A(15000)
REAL*4 B(I)
EQUIVALENCE (A(1), B(I))

C DEFINE THE COMMON BLOCKS FOR INPUT AND OUTPUT TO THE INTERPOLATION
C SUBROUTINE, NOTE THE DENSITY AND TEMPERATURE MUST BE INPUT AS
C DOUBLE PRECISION
REAL*8 X, V
COMMON/SESIN/ IR, IDT, X, Y, IBR, IFL, IH
COMMON/SESOUT/ Z(3), ZZ(3)
COMMON/INTORD/ IFN
COMMON/RTBLK2/ LOCK, IX, NX, LOCY, IV

C SET UP THE MATERIAL ID'S FOR BERYLLIUM
MATS(1,1) = 2020
MATS(2,1) = 12020

C DEFINE THE IO UNIT NUMBERS
16 = 16
LES = 3
LOP = 3
LOUT = 16

C SET THE LONG OUTPUT SWITCH TO ON
LONGIO = 1

C DEFINE THE LENGTH OF THE VECTOR TO HOLD THE SESAME DATA AND SET THE
C POINTER TO THE FIRST ELEMENT
LEN A = 15000
LCNT = 1

C SET THE OPACITY OPTION TO GET OPACITIES AND SET THE INTERPOLATION
C OPTION TO USE THE RATIONAL FUNCTION APPROXIMATION
IRAD = 1
IFN = 0

C SET THE UNIT CONVERSION FACTORS TO SI UNITS
C G/CM3 --> KG/M3
C K --> K
C GJ/M3 --> PA
C GJ/MG --> J/KG

UCONV(1,1) = 1.E3
UCONV(2,1) = 1.E0
UCONV(3,1) = 1.E9
UCONV(3,2) = 1.E6

C G/CM3 --> KG/M3
C EV --> K
C -- --> --

UCONV(1,3) = 1.E3
UCONV(2,3) = 11604.E0
UCONV(3,3) = 1.E0

C G/CM3 --> KG/M3
C EV --> K
C CM2/G --> M2/KG

UCONV(1,4) = 1.E3
UCONV(2,4) = 11604.E0
UCONV(3,4) = 0.1E0
UCONV(3,5) = 0.1E0

C RETRIEVE THE SESAME DATA
C CALL GETLIB(LEN A, LCNT, NMATS, MATS, NWDESE, UCONV,
C IRAD, LES, LOP, LOUT, LONGIO)

C INTERPOLATE FOR 20 (DENSITY, TEMPERATURE) POINTS

D(1) = 1.845E-2
D(2) = 1.845E-1
D(3) = 1.845
D(4) = 18.45
T(C) = 11604.E-2
T(5) = 11604.E-1
T(9) = 11604.
T(13) = 11604.E1
T(17) = 11604.E2

KMAX = 20
 DO 100 K = 1,KMAX
.IMAT(K) = 1
100 CONTINUE
DO 150 K = 1,KMAX,4
D(4+K) = D(1)
D(5+K) = D(2)
D(6+K) = D(3)
D(7+K) = D(4)
150 CONTINUE

DO 200 K = 1,3
T(I+K) = T(I)
T(5+K) = T(5)
T(9+K) = T(9)
T(13+K) = T(13)
T(17+K) = T(17)
200 CONTINUE

CALL EOSTAB(IMAT,T,D,P,DPDR,DPDT,E,DED,R,DED,T, 1  Z,SR,SP,XLOG,VLOG,KMAX,IRAD)

WRITE(16,3000) (I, D(I), T(I), P(I), DPDR(I), DPDT(I), 1  E(I), DEDR(I), DEDT(I), 1  Z(I), SR(I), SP(I), 1
  I=1,KMAX)

3000 FORMAT(1X,'DENSITY TEMPER PRESSURE DP/DD DP/DT ', 1  'ENERGY DE/DD DE/DT ', 1  'CHARGE ROSS OP PLANCK OP'/ 1 (1X,I3,1P11E10.3))

STOP
END
### Material Data Table

**Index**

<table>
<thead>
<tr>
<th>P &amp; E</th>
<th>Z &amp; OPAC</th>
<th>Densities Temps</th>
<th>Pointer</th>
<th>Densities Temps</th>
<th>Pointer</th>
<th>Densities Temps</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2020</td>
<td>12020</td>
<td>101</td>
<td>23</td>
<td>2452</td>
<td>31</td>
</tr>
</tbody>
</table>

**Material List**

<table>
<thead>
<tr>
<th>Material ID</th>
<th>Density</th>
<th>Temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000E+00</td>
<td>1.441E+01</td>
<td>2.750E+01</td>
</tr>
<tr>
<td>4.059E+01</td>
<td>7.795E+01</td>
<td>1.153E+02</td>
</tr>
<tr>
<td>1.906E+02</td>
<td>2.660E+02</td>
<td>3.413E+02</td>
</tr>
<tr>
<td>4.981E+02</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Data Type**

- 1 = P & E, 2 = Z, 3 = OP

**Number of Densities Points**

- 101

**Number of Temperature Points**

- 23

**Long Output of Material Data Table**

- TEMPE \ DENPE
- 0.000E+00 1.441E+01 2.750E+01 4.059E+01 7.795E+01 1.153E+02 1.906E+02 2.660E+02 3.413E+02 4.981E+02

**Additional Notes**

- Long output for materials with different densities and temperatures is provided, detailing specific values for each entry.

**Additional Table**

<table>
<thead>
<tr>
<th>Temp \ DenPE</th>
<th>6.458E+02</th>
<th>1.841E+02</th>
<th>9.225E+02</th>
<th>1.031E+03</th>
<th>1.153E+03</th>
<th>1.281E+03</th>
<th>1.347E+03</th>
<th>1.476E+03</th>
<th>1.591E+03</th>
<th>1.693E+03</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>2.901E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>5.802E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>1.160E+03</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>2.321E+04</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>TEMP \ DEN</td>
<td>3.604E+03</td>
<td>3.690E+03</td>
<td>3.812E+03</td>
<td>3.926E+03</td>
<td>3.955E+03</td>
<td>4.048E+03</td>
<td>4.174E+03</td>
<td>4.311E+03</td>
<td>4.457E+03</td>
<td>4.613E+03</td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
</tr>
<tr>
<td>2.566E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
</tr>
<tr>
<td>2.566E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
</tr>
<tr>
<td>2.566E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
</tr>
<tr>
<td>2.566E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
</tr>
<tr>
<td>2.566E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
</tr>
<tr>
<td>2.566E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
</tr>
<tr>
<td>2.566E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
</tr>
<tr>
<td>2.566E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
</tr>
<tr>
<td>2.566E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
</tr>
<tr>
<td>2.566E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
</tr>
<tr>
<td>2.566E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
</tr>
<tr>
<td>2.566E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
</tr>
<tr>
<td>2.566E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
</tr>
<tr>
<td>2.566E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
<td>2.556E+00</td>
</tr>
<tr>
<td>TEMP \ DEN</td>
<td>1.000E-03</td>
<td>2.150E-03</td>
<td>4.640E-03</td>
<td>1.000E-01</td>
<td>2.150E-01</td>
<td>4.640E-01</td>
<td>1.000E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.450E+04</td>
<td>9.011E-00</td>
<td>8.974E-00</td>
<td>8.001E-00</td>
<td>6.845E-00</td>
<td>5.689E-00</td>
<td>4.632E-00</td>
<td>3.676E-00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.741E+04</td>
<td>1.232E+00</td>
<td>1.127E+00</td>
<td>1.050E+00</td>
<td>9.957E+00</td>
<td>9.503E+00</td>
<td>8.979E+00</td>
<td>8.316E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.321E+04</td>
<td>1.831E+00</td>
<td>1.726E+00</td>
<td>1.659E+00</td>
<td>1.644E+00</td>
<td>1.602E+00</td>
<td>1.560E+00</td>
<td>1.522E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.481E+04</td>
<td>1.970E+00</td>
<td>1.956E+00</td>
<td>1.940E+00</td>
<td>1.910E+00</td>
<td>1.861E+00</td>
<td>1.815E+00</td>
<td>1.772E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.642E+04</td>
<td>1.979E+00</td>
<td>1.973E+00</td>
<td>1.966E+00</td>
<td>1.956E+00</td>
<td>1.943E+00</td>
<td>1.924E+00</td>
<td>1.881E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.802E+04</td>
<td>1.981E+00</td>
<td>1.978E+00</td>
<td>1.975E+00</td>
<td>1.971E+00</td>
<td>1.968E+00</td>
<td>1.955E+00</td>
<td>1.946E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.962E+04</td>
<td>1.983E+00</td>
<td>1.980E+00</td>
<td>1.978E+00</td>
<td>1.974E+00</td>
<td>1.970E+00</td>
<td>1.955E+00</td>
<td>1.948E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.283E+04</td>
<td>1.996E+00</td>
<td>1.988E+00</td>
<td>1.982E+00</td>
<td>1.978E+00</td>
<td>1.975E+00</td>
<td>1.970E+00</td>
<td>1.961E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.160E+05</td>
<td>2.331E+00</td>
<td>2.191E+00</td>
<td>2.095E+00</td>
<td>2.039E+00</td>
<td>2.005E+00</td>
<td>1.987E+00</td>
<td>1.970E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.450E+05</td>
<td>2.930E+00</td>
<td>2.850E+00</td>
<td>2.731E+00</td>
<td>2.575E+00</td>
<td>2.405E+00</td>
<td>2.249E+00</td>
<td>2.133E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.741E+05</td>
<td>3.268E+00</td>
<td>3.133E+00</td>
<td>3.033E+00</td>
<td>2.952E+00</td>
<td>2.862E+00</td>
<td>2.741E+00</td>
<td>2.609E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.321E+05</td>
<td>3.937E+00</td>
<td>3.868E+00</td>
<td>3.798E+00</td>
<td>3.726E+00</td>
<td>3.659E+00</td>
<td>3.584E+00</td>
<td>3.512E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.481E+05</td>
<td>3.984E+00</td>
<td>3.976E+00</td>
<td>3.963E+00</td>
<td>3.940E+00</td>
<td>3.925E+00</td>
<td>3.913E+00</td>
<td>3.902E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.642E+05</td>
<td>3.990E+00</td>
<td>3.985E+00</td>
<td>3.979E+00</td>
<td>3.973E+00</td>
<td>3.962E+00</td>
<td>3.941E+00</td>
<td>3.908E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.802E+05</td>
<td>3.994E+00</td>
<td>3.991E+00</td>
<td>3.986E+00</td>
<td>3.982E+00</td>
<td>3.974E+00</td>
<td>3.961E+00</td>
<td>3.952E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.962E+05</td>
<td>3.995E+00</td>
<td>3.993E+00</td>
<td>3.990E+00</td>
<td>3.986E+00</td>
<td>3.983E+00</td>
<td>3.979E+00</td>
<td>3.971E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.283E+05</td>
<td>3.997E+00</td>
<td>3.995E+00</td>
<td>3.993E+00</td>
<td>3.990E+00</td>
<td>3.986E+00</td>
<td>3.982E+00</td>
<td>3.975E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.160E+06</td>
<td>3.996E+00</td>
<td>3.997E+00</td>
<td>3.996E+00</td>
<td>3.995E+00</td>
<td>3.992E+00</td>
<td>3.990E+00</td>
<td>3.981E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.450E+06</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.741E+06</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.321E+06</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.481E+06</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.802E+06</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.962E+06</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.283E+06</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.160E+07</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.321E+07</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td>4.000E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEMP</td>
<td>DEN</td>
<td>1.000E+07</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td>-----------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.160E+04</td>
<td>9.109E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.450E+04</td>
<td>1.001E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.741E+04</td>
<td>1.232E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.321E+04</td>
<td>1.831E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.901E+04</td>
<td>1.951E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.481E+04</td>
<td>1.970E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.642E+04</td>
<td>1.979E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.802E+04</td>
<td>1.981E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.962E+04</td>
<td>1.983E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.283E+04</td>
<td>1.997E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.160E+05</td>
<td>2.331E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.450E+05</td>
<td>2.930E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.741E+05</td>
<td>3.268E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.321E+05</td>
<td>3.937E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.901E+05</td>
<td>3.984E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.481E+05</td>
<td>3.990E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.642E+05</td>
<td>3.994E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.802E+05</td>
<td>3.995E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.962E+05</td>
<td>3.996E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.283E+05</td>
<td>3.998E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.160E+06</td>
<td>3.998E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.450E+06</td>
<td>4.000E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.741E+06</td>
<td>4.000E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.321E+06</td>
<td>4.000E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.901E+06</td>
<td>4.000E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.481E+06</td>
<td>4.000E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.642E+06</td>
<td>4.000E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.802E+06</td>
<td>4.000E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.962E+06</td>
<td>4.000E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.283E+06</td>
<td>4.000E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.160E+07</td>
<td>4.000E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.160E+09</td>
<td>4.000E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**LONG OUTPUT OF SESAME DATA TABLE**

<table>
<thead>
<tr>
<th>SESAME MATERIAL ID.</th>
<th>12020</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA TYPE</td>
<td>1=P&amp;E,2=Z,3=OP.</td>
</tr>
<tr>
<td>NUMBER OF DENSITY POINTS</td>
<td>31</td>
</tr>
<tr>
<td>NUMBER OF TEMPERATURE POINTS</td>
<td>46</td>
</tr>
<tr>
<td>POINTER INTO TBL VECTOR</td>
<td>3511</td>
</tr>
<tr>
<td>LEFT=1, RIGHT=2 PACKED DATA ELEMENT</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TEMP</th>
<th>DEN</th>
<th>1.000E+03</th>
<th>2.150E+03</th>
<th>4.640E+03</th>
<th>1.000E+02</th>
<th>2.150E+02</th>
<th>4.640E+02</th>
<th>1.000E+01</th>
<th>2.150E+01</th>
<th>4.640E+01</th>
<th>1.000E+00</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.160E+04</td>
<td>1.972E+02</td>
<td>4.222E+02</td>
<td>8.475E+02</td>
<td>1.528E+03</td>
<td>2.478E+03</td>
<td>3.715E+03</td>
<td>5.284E+03</td>
<td>7.259E+03</td>
<td>9.555E+03</td>
<td>1.198E+04</td>
<td></td>
</tr>
<tr>
<td>DENSITY</td>
<td>TEMPER</td>
<td>PRESSURE</td>
<td>DP/DD</td>
<td>DP/DT</td>
<td>ENERGY</td>
<td>DE/DD</td>
<td>DE/DT</td>
<td>CHARGE</td>
<td>ROSS OP</td>
<td>PLANCK OP</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>----------</td>
<td>-------</td>
<td>-------</td>
<td>--------</td>
<td>-------</td>
<td>-------</td>
<td>--------</td>
<td>---------</td>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td>1.84E-02</td>
<td>1.16E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>2.634E+05</td>
<td>0.000E+00</td>
<td>6.089E+02</td>
<td>1.935E-31</td>
<td>7.11E-31</td>
<td>1.528E+26</td>
<td></td>
</tr>
<tr>
<td>1.84E-01</td>
<td>1.16E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>2.634E+05</td>
<td>0.000E+00</td>
<td>6.089E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>3.485E-25</td>
<td></td>
</tr>
<tr>
<td>1.84E-00</td>
<td>1.16E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>2.634E+05</td>
<td>0.000E+00</td>
<td>6.089E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td></td>
</tr>
<tr>
<td>1.84E-02</td>
<td>1.16E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>2.634E+05</td>
<td>0.000E+00</td>
<td>6.089E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td></td>
</tr>
<tr>
<td>1.84E-1</td>
<td>1.16E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>2.634E+05</td>
<td>0.000E+00</td>
<td>6.089E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td></td>
</tr>
<tr>
<td>1.84E-02</td>
<td>1.16E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>2.634E+05</td>
<td>0.000E+00</td>
<td>6.089E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td></td>
</tr>
<tr>
<td>1.84E-1</td>
<td>1.16E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>2.634E+05</td>
<td>0.000E+00</td>
<td>6.089E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td></td>
</tr>
<tr>
<td>1.84E-02</td>
<td>1.16E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>2.634E+05</td>
<td>0.000E+00</td>
<td>6.089E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td></td>
</tr>
<tr>
<td>1.84E-1</td>
<td>1.16E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>2.634E+05</td>
<td>0.000E+00</td>
<td>6.089E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td></td>
</tr>
<tr>
<td>1.84E-02</td>
<td>1.16E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>2.634E+05</td>
<td>0.000E+00</td>
<td>6.089E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td></td>
</tr>
<tr>
<td>1.84E-1</td>
<td>1.16E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>2.634E+05</td>
<td>0.000E+00</td>
<td>6.089E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td></td>
</tr>
<tr>
<td>1.84E-02</td>
<td>1.16E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>2.634E+05</td>
<td>0.000E+00</td>
<td>6.089E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td></td>
</tr>
<tr>
<td>1.84E-1</td>
<td>1.16E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>2.634E+05</td>
<td>0.000E+00</td>
<td>6.089E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td></td>
</tr>
<tr>
<td>1.84E-02</td>
<td>1.16E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>2.634E+05</td>
<td>0.000E+00</td>
<td>6.089E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td></td>
</tr>
</tbody>
</table>

INTERPOLATION OUTPUT
<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>1.845E+01</td>
<td>1.160E+05</td>
<td>5.252E+09</td>
<td>2.605E+08</td>
<td>6.322E+04</td>
<td>1.023E+08</td>
<td>8.260E+06</td>
<td>1.145E+04</td>
</tr>
<tr>
<td>17</td>
<td>1.845E-02</td>
<td>1.160E+06</td>
<td>9.491E+07</td>
<td>5.144E+09</td>
<td>8.628E+01</td>
<td>1.127E+10</td>
<td>0.000E+00</td>
<td>8.295E+03</td>
</tr>
<tr>
<td>18</td>
<td>1.845E-01</td>
<td>1.160E+06</td>
<td>9.489E+08</td>
<td>5.143E+09</td>
<td>8.628E+02</td>
<td>1.127E+10</td>
<td>0.000E+00</td>
<td>8.295E+03</td>
</tr>
<tr>
<td>19</td>
<td>1.845E+00</td>
<td>1.160E+06</td>
<td>9.479E+09</td>
<td>5.131E+09</td>
<td>8.632E+03</td>
<td>1.127E+10</td>
<td>0.000E+00</td>
<td>8.295E+03</td>
</tr>
<tr>
<td>20</td>
<td>1.845E+01</td>
<td>1.160E+06</td>
<td>9.368E+10</td>
<td>5.003E+09</td>
<td>8.666E+04</td>
<td>1.120E+10</td>
<td>-2.176E+07</td>
<td>8.307E+03</td>
</tr>
</tbody>
</table>

***EOF***

22 PAGES COST $3.30  PROJ BALANCE $10538.95  USER BALANCE $425.24
C A PROGRAM TO TEST THE TOTAL, NUCLEAR, AND ELECTRON EQUATION OF STATE
C TABLES FOR CONSISTENCY

REAL*8 A
COMMON /SESDAT/ A(40000)
DIMENSION UCONV(3,5), P(5000), E(5000)
INCLUDE 'SESPAR.CMN'
COMMON /SZDIR/ LCMX, NRS, LCFW(NREG,NDAT)
REAL*8 DDATA, DDATA2, DDATA3
DIMENSION SDATA(2), SDATA2(2), SDATA3(2)
EQUIVALENCE (DDATA, SDATA(1))
EQUIVALENCE (DDATA2, SDATA2(1))
EQUIVALENCE (DDATA3, SDATA3(1))

C SET THE INPUT TO THE GET... ROUTINES

LCMX = 40000
NRS = NREG
IR = 1
MID = 2020
LCNT = 1
LU = 3
LOUT = 4

C CONVERT TO CGS UNITS

UCONV(1,1) = 1.
UCONV(2,1) = 1.
UCONV(3,1) = 1.E9
UCONV(3,2) = 1.E10

C GET THE TOTAL PRESSURE AND SPECIFIC INTERNAL ENERGY

IDT = 1
CALL GETEOS(IR,MID,IDT,A,LCNT,LU,IFL,ZB,UCONV)

C GET THE NUCLEAR PRESSURE AND SPECIFIC INTERNAL ENERGY

IDT = 2
CALL GETNUC(IR,MID,IDT,A,LCNT,LU,IFL,UCONV)

C GET THE ELECTRON PRESSURE AND SPECIFIC INTERNAL ENERGY

IDT = 3
CALL GETELC(IR,MID,IDT,A,LCNT,LU,IFL,UCONV)

C GET POINTERS TO THE LOADED TABLES

LOC1 = LCFW(1,1)
LOC2 = LCFW(1,2)
LOC3 = LCFW(1,3)

C COMBINE THE NUCLEAR AND ELECTRON VALUES

NR = A(LOC1+2)
NT = A(LOC1+3)
NRNT = NR * NT
K = 0
DO 10 J = 1,NT
10 DO 1 I = 1, NR
K = K + 1
DDATA = A(LOC2+3+NR+NT+K)
P(K) = SDATA(1)
E(K) = SDATA(2)
DDATA = A(LOC3+3+NR+NT+K)
P(K) = P(K) + SDATA(1)
E(K) = E(K) + SDATA(2)
10 CONTINUE

C PRINT RESULTS

K = 0
WRITE(LOUT,1000)
DO 20 J = 1,NT
DO 20 I = 1, NR
K = K + 1
DDATA = A(LOC1+3+NR+NT+K)
DDATA2 = A(LOC2+3+NR+NT+K)
DDATA3 = A(LOC3+3+NR+NT+K)
WRITE(LOUT,1001) K, A(LOC1+3+I), A(LOC1+3+NR+J),
1 SDATA(1), SDATA(2),
2 P(K), E(K),
3 SDATA2(1), SDATA2(2),
4 SDATA3(1), SDATA3(2)
20 CONTINUE

STOP

1000 FORMAT(12X,
1 ' DENSITY   TEMPERATURE  TOT PRESS  TOT ENERGY  COMB PRESS ','
2 ' COMB ENERGY  NUCL PRESS  NUCL ENERGY  ELEC PRESS  ELEC ENERGY'/',
3 ' 0')
1001 FORMAT(I12,1P10E12.4)
END
| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

Table 1: Example Table

<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
<th>Column 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data 1</td>
<td>Data 2</td>
<td>Data 3</td>
<td>Data 4</td>
</tr>
<tr>
<td>Data 5</td>
<td>Data 6</td>
<td>Data 7</td>
<td>Data 8</td>
</tr>
<tr>
<td>Data 9</td>
<td>Data 10</td>
<td>Data 11</td>
<td>Data 12</td>
</tr>
</tbody>
</table>

...
<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2316</td>
<td>1.1282E+03</td>
<td>3.7136E+08</td>
<td>1.9336E+18</td>
<td>2.5872E+16</td>
<td>1.9336E+18</td>
<td>2.5872E+16</td>
<td>4.6748E+17</td>
</tr>
<tr>
<td>2317</td>
<td>1.3422E+03</td>
<td>3.7136E+08</td>
<td>2.3025E+18</td>
<td>2.5907E+16</td>
<td>2.3025E+18</td>
<td>2.5907E+16</td>
<td>5.6269E+17</td>
</tr>
<tr>
<td>2318</td>
<td>1.6439E+03</td>
<td>3.7136E+08</td>
<td>2.8237E+18</td>
<td>2.5954E+16</td>
<td>2.8237E+18</td>
<td>2.5954E+16</td>
<td>6.9948E+17</td>
</tr>
</tbody>
</table>
## Master Directory

<table>
<thead>
<tr>
<th>Record #</th>
<th>Word #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Number of materials in library (NMAT)</td>
</tr>
<tr>
<td></td>
<td>2-3</td>
<td>Date and version number</td>
</tr>
<tr>
<td>2</td>
<td>1+NMAT</td>
<td>Material ID numbers</td>
</tr>
<tr>
<td></td>
<td>NMAT+1+2*NMAT</td>
<td>Length of material index directories</td>
</tr>
</tbody>
</table>

## Material Index Directory

<table>
<thead>
<tr>
<th>Record #</th>
<th>Word #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Material ID number</td>
</tr>
<tr>
<td></td>
<td>2-4</td>
<td>Miscellaneous</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>Number of data tables for this material (NTBLS)</td>
</tr>
<tr>
<td></td>
<td>6-6+NTBLS</td>
<td>Catalog numbers of data tables</td>
</tr>
<tr>
<td></td>
<td>7+NTBLS+7+2*NTBLS</td>
<td>Lengths of data tables</td>
</tr>
</tbody>
</table>

## Material Data Tables

See Tables 3.4 to 3.7

---

**Fig. 3.4. Format of Data in the SESAME Library**
Table 3.3. Units in the SESAME Binary Library

**Equation of State Library**

<table>
<thead>
<tr>
<th>Property</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density</td>
<td>grams/cm³</td>
</tr>
<tr>
<td>Temperature</td>
<td>Kelvin</td>
</tr>
<tr>
<td>Pressure</td>
<td>GJ/m³</td>
</tr>
<tr>
<td>Specific Energy</td>
<td>GJ/MG</td>
</tr>
</tbody>
</table>

GJ = 10⁹ J  
MG = 10⁶ grams

**Astrophysical Sublibrary***

<table>
<thead>
<tr>
<th>Property</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density</td>
<td>grams/cm³</td>
</tr>
<tr>
<td>Temperature</td>
<td>eV</td>
</tr>
<tr>
<td>Charge</td>
<td>---</td>
</tr>
<tr>
<td>Opacity</td>
<td>cm²/gram</td>
</tr>
</tbody>
</table>

**Conductivity Sublibrary***

<table>
<thead>
<tr>
<th>Property</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density</td>
<td>grams/cm³</td>
</tr>
<tr>
<td>Temperature</td>
<td>eV</td>
</tr>
<tr>
<td>Electrical Conductivity</td>
<td>sec⁻¹</td>
</tr>
<tr>
<td>Thermal Conductivity</td>
<td>sec⁻¹ cm⁻¹</td>
</tr>
<tr>
<td>Thermoelectric Coefficient</td>
<td>sec⁻¹ cm⁻¹</td>
</tr>
</tbody>
</table>

*All data is stored as log₁₀.
<table>
<thead>
<tr>
<th>Record</th>
<th>Word</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Mean atomic number</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Mean atomic mass</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Normal density</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Normal bulk modulus</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>Exchange coefficient</td>
</tr>
</tbody>
</table>
total pressure and specific energy are both stored in the 301. table for each material in the SESAME binary library. The format of this data is given in Table 3.5 where each data element is a 32-bit word. This is converted by GETEOS into the format given in Table 3.5 where each data element is now a 64-bit double precision variable. This is done to allow easy packing and unpacking of the double packed pressure/energy data. To unpack a table element we use the EQUIVALENCE of double and single precision variables.

\[
\begin{align*}
\text{REAL*8 DDATA} \\
\text{REAL*4 SDATA(2)} \\
\text{EQUIVALENCE (DDATA,SDATA(1))} \\
\vdots \\
\text{DDATA=TBLS(...)} \\
\text{P=SDATA(1)} \\
\text{E=SDATA(2)} \\
\vdots
\end{align*}
\]

This is standard SESAME format because the original coding is for double packing two short numbers into 60 bit CDC 7600 words. By converting the data tables to IBM double precision we have made the coding more portable to 32-bit word length computers and do not require special masking operations or bit-shift operations that are different for all compilers. We make extensive use of automatic type conversion in the routines. For instance the number of densities in the table is double precision floating point and we convert to integer using

\[
\text{NR = TBLS(LOC + 2)}.\]

### Table 3.5. Format of 301. Table in SESAME Binary Library

<table>
<thead>
<tr>
<th>Record</th>
<th>Word</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>number of densities (NR)</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>number of temperatures (NT)</td>
</tr>
<tr>
<td>3</td>
<td>3+NR</td>
<td>list of densities</td>
</tr>
<tr>
<td>4+NR</td>
<td>4+NR+NT</td>
<td>list of temperatures</td>
</tr>
<tr>
<td>5+NR+NT</td>
<td>5+NR+NT+NR*NT</td>
<td>pressure data (density index varies fastest)</td>
</tr>
<tr>
<td>6+NR+NT+NR*NT</td>
<td>6+NR+NT+2<em>NR</em>NT</td>
<td>specific energy data</td>
</tr>
</tbody>
</table>

#### Format of Pressure and Energy Data Table in TBLS

<table>
<thead>
<tr>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBLS(LOC)</td>
<td>Material ID</td>
</tr>
<tr>
<td>TBLS(LOC+1)</td>
<td>Solid density of material</td>
</tr>
<tr>
<td>TBLS(LOC+2)</td>
<td>Number of density mesh points (NR)</td>
</tr>
<tr>
<td>TBLS(LOC+3)</td>
<td>Number of temperature mesh points (NT)</td>
</tr>
<tr>
<td>TBLS(LOC+4)</td>
<td>Vector of densities</td>
</tr>
<tr>
<td>TBLS(LOC+4+NR)</td>
<td>Vector of temperature</td>
</tr>
<tr>
<td>TBLS(LOC+4+NR+NT)</td>
<td>Matrix of packed pressure/energy data (density varying fastest)</td>
</tr>
</tbody>
</table>
These densities could be printed using

\[ \text{WRITE}(6,*)\text{(TBLS(LOC + 3 + I), I = 1, NR)}. \]

We could also have used

\[ \text{WRITE}(6,*)\text{(TBLS(LOC + 3 + I), I=1, INT(TLBS(LOC + 2))}. \]

3.1.2.2 GETNFE -- Number of Free Electron Data

The number of free electrons per atom (called the charge state) is stored in the 504. table for each material. This table has the format shown in Table 3.6 in the SESAME binary library. It is converted by the GETNFE routine into the format shown in Table 3.6 where the data is packed into the left half of the double word leaving the right half as zero. This is done so the interpolation routines can be used on it without modification from their double packed implementation. This second half could be used for additional data such as electric or thermal conductivity for instance.

The density, temperature, and charge state data are all stored in the SESAME binary library as \( \log_{10} \) of their actual values. This is true for all data in the astrophysical sub-library and is distinctly different than the data in the equation of state sub-library. Furthermore, the units used in the astrophysical sub-library are different than those used in the equation of state sub-library as noted in Table 3.3. Unit conversions in GETNFE are done using additive factors because the data is logarithmic:

\[ \rho^*f + \log_{10} \rho + \log_{10} f. \]
Table 3.6. Format of 504. Table in SESAME Binary Library

<table>
<thead>
<tr>
<th>Record</th>
<th>Word</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>number of densities (NR)</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>number of temperatures (NT)</td>
</tr>
<tr>
<td>3 + 3+NR</td>
<td></td>
<td>list of log&lt;sub&gt;10&lt;/sub&gt; densities</td>
</tr>
<tr>
<td>4+NR + 4+NR+NT</td>
<td></td>
<td>list of log&lt;sub&gt;10&lt;/sub&gt; temperatures</td>
</tr>
<tr>
<td>5+NR+NT + 5+NR+NT+NR*NT</td>
<td></td>
<td>charge state data (density index varies fastest)</td>
</tr>
</tbody>
</table>

Format of Charge State Data Table in TBLS

<table>
<thead>
<tr>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBLS(LOC)</td>
<td>Material ID</td>
</tr>
<tr>
<td>TBLS(LOC+1)</td>
<td>Data type IDT=2</td>
</tr>
<tr>
<td>TBLS(LOC+2)</td>
<td>Number of density mesh points (NR)</td>
</tr>
<tr>
<td>TBLS(LOC+3)</td>
<td>Number of temperature mesh points (NT)</td>
</tr>
<tr>
<td>TBLS(LOC+4)</td>
<td>Vector of log&lt;sub&gt;10&lt;/sub&gt; densities</td>
</tr>
<tr>
<td>TBLS(LOC+4+NR)</td>
<td>Vector of log&lt;sub&gt;10&lt;/sub&gt; temperatures</td>
</tr>
<tr>
<td>TBLS(LOC+4+NR+NT)</td>
<td>Matrix of packed log&lt;sub&gt;10&lt;/sub&gt; charge state data (density varies fastest)</td>
</tr>
</tbody>
</table>
3.1.2.3 GETRPO -- Rosseland and Planck Opacity Data

The Rosseland opacity data is stored in table 502. for each material and the Planck opacity data is stored in table 505. Each of these has the format shown in Table 3.7. They are double-packed by GETRPO into the format shown in Table 3.7. The data is all log_{10} of the actual values. Hence this is again different than either the pressure and energy data and the charge state data. We have maintained this inconsistency because it is standard SESAME implementation and we believe that adherence to the standard is important for compatibility with future SESAME releases.

3.1.3 Other "GET..." Subroutines Implemented in KfK-INR SESAME

There are 4 other GET... subroutines included in the SESAME package from LANL. Those that are included in the KfK-INR SESAME package are discussed here. These routines, like the GETEOS, GETNFE and GETRPO routines, can be called directly by an application program if desired.

3.1.3.1 GETNUC -- Ion Equation of State Data

This subroutine works in the same manner as GETEOS except that GETNUC retrieves the 303. table that contains the pressure and specific internal energy data for the ion component of a plasma. The GETNUC routine uses the same unit conversion factors as GETEOS:

\[ RFAC = UCONV(1,1) \]
\[ TFAC = UCONV(2,1) \]
\[ PFAC = UCONV(3,1) \]
\[ EFAC = UCONV(3,2). \]

3.1.3.2 GETELC -- Electron Equation of State Data

This subroutine works in the same manner as GETEOS except that GETELC retrieves the 304. table that contains the pressure and specific internal energy
Table 3.7. Format of 502. and 505. Tables in SESAME Binary Library

<table>
<thead>
<tr>
<th>Record</th>
<th>Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3 → 3+NR</td>
<td></td>
</tr>
<tr>
<td>4+NR → 4+NR+NT</td>
<td></td>
</tr>
<tr>
<td>5+NR+NT → 5+NR+NT+NR*NT</td>
<td></td>
</tr>
</tbody>
</table>

Format of Opacity Data Table in TBLS

<table>
<thead>
<tr>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBLS(LOC)</td>
<td>Material ID</td>
</tr>
<tr>
<td>TBLS(LOC+1)</td>
<td>Data type IDT=3</td>
</tr>
<tr>
<td>TBLS(LOC+2)</td>
<td>Number of density mesh points (NR)</td>
</tr>
<tr>
<td>TBLS(LOC+3)</td>
<td>Number of temperature mesh points (NT)</td>
</tr>
<tr>
<td>TBLS(LOC+4)</td>
<td>Vector of log_{10} densities</td>
</tr>
<tr>
<td>TBLS(LOC+4+NR)</td>
<td>Vector of log_{10} temperatures</td>
</tr>
<tr>
<td>TBLS(LOC+4+NR+NT)</td>
<td>Matrix of packed log_{10} Rosseland/Planck opacity data (density varies fastest)</td>
</tr>
</tbody>
</table>
of the electron component of a plasma. The GETELC routine was created by FPA. No GETELC routine was included in the SESAME package from LANL. The GETELC routine uses the same unit conversion factors as GETNUC.

3.1.3.3 GETTCE -- Thermal Conductivity Data

The GETTCE subroutine retrieves the 503. table that contains the electron thermal conductivity. This routine was only modified to use the single/double precision equivalencing to pack the data into the left half of double words. Unit conversions are the same as in the original SESAME package and must be modified to suit the user's application.

3.1.3.4 GETRTL -- Total Opacity Data

The GETRTL routine retrieves the Rosseland and total radiation plus electron opacity and double packs these into double words. It uses the unit conversion factors:

\[
\begin{align*}
RFAC &= UCONV(1,4) \\
TFAC &= UCONV(2,4) \\
ROFAC &= UCONV(3,4) \\
TOFAC &= UCONV(3,5).
\end{align*}
\]

3.1.4 GET... Subroutines Not Implemented in KfK-INR SESAME

The GETINV subroutine was not implemented. This routine creates an "inverted" equation of state table using the 301 table as its starting point. In the inverted table they use density and specific internal energy as independent variables and pressure and temperature as dependent variables. This routine did not work properly as it was received from LANL. Two days were spent in a debugging effort with no success. Since we believe that this routine is not currently needed by the KfK codes using SESAME we dropped the effort to convert it.
4. Interpolation in Data Tables

4.1 SESAME Interpolation Routines

The interpolation routines supplied with the SESAME library are T4DAT which in turn calls T4INTP. These routines are not vectorized and return only values and derivatives for a single (density, temperature) point and a single data type. However, they are smart enough to remember the last call to them and this speeds the interpolation of the next call if the same material and data type are specified. Hence they should be used in the following manner:

FOR ALL ZONES OF MATERIAL N
CALL T4DAT FOR P&E

FOR ALL ZONES OF MATERIAL N
CALL T4DAT FOR Z

FOR ALL ZONES OF MATERIAL N
CALL T4DAT FOR OPACITIES.

Indexing through all zones sequentially will generally accomplish this since materials are usually grouped together in adjoining zones.

The T4DAT subroutine finds the indices in the density and temperature vectors that are closest to the input values of density and temperature. If the material and data type are the same as in the last call to T4DAT then it starts the search at the values from the last call. Since hydrodynamics codes solve for continuous density and temperature functions there is great likelihood that adjoining mesh points will have nearly the same temperature and
density. This feature speeds the search process. Otherwise a binary search is done, starting at the center of the density and temperature vectors.

Once the density and temperature indices are found, T4INTP is called to compute the interpolation coefficients and to compute the interpolated value. If the material type, data category and temperature and density indices are the same as in the previous call then the coefficients are not recomputed and the interpolation is done with saved coefficients. Again, this speeds the calculation.

Unlike the GET... routines, all communication with the interpolation routines is through COMMON blocks. Table 4.1 contains descriptions of the COMMON blocks used in the interpolation routines.

4.2 Subroutine EOSTAB

A routine called EOSTAB has been written by FPA to implement calls to T4DAT to retrieve equation of state, charge state and opacity data for all zones of a calculational mesh. The argument list for EOSTAB is given in Table 4.2. The example calculation given in Fig. 3.3 demonstrates the cell to EOSTAB.

4.3 KfK Interpolation Routines

The KfK interpolation routine LOOKUP evaluates data for vectors of density and temperature using a fast lookup method that is hardware-specific to IBM 360 floating point data format. To use this method, additional information about the data must be saved. This is done by calling INILOK from GETLIB and storing the additional information in COMMON/KFKDAT/ for subsequent calls to LOOKUP. These statements have been "commented out" of the source code but can be activated by removing the comment character "C" from column 1.
Table 4.1. Common Blocks Used by SESAME Interpolation Subroutines

COMMON/SESIN/

IDT -- Data category = 1 P&E, = 2 Z, = 3 Opacities
AR -- REAL*8 value of density
AT -- REAL*8 value of temperature
IBR -- Selector for left, right, or both double packed data values
       = 0 both, = 1 left, = 2 right

COMMON/SESOUT/

Z1(3) -- Z1(1) = left interpolated data value
        Z1(2) = ∂Z1(1)/∂p
        Z1(3) = ∂Z1(1)/∂T

Z2(3) -- Z2(1) = right interpolated data value
        Z2(2) = ∂Z2(1)/∂p
        Z3(2) = ∂Z2(1)/∂T

COMMON/INTORD/

IFN -- Selector of interpolation scheme
       = 0 rational function interpolation
       = 1 bilinear interpolation

COMMON/RTBLK2/

LOCX R
IX T
NX Z(3)
LOC4 INT
I4 IDS Indexes and values used in the interpolation process
N4 ZZ(ICDIM)
LOCZ IHH
NZ LOCH
NSFT NH
Table 4.2. Argument List for Subroutine EOSTAB

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMAT</td>
<td>Vector of material region numbers (Input)</td>
</tr>
<tr>
<td>T</td>
<td>Vector of temperatures (Input)</td>
</tr>
<tr>
<td>D</td>
<td>Vector of densities (Input)</td>
</tr>
<tr>
<td>P</td>
<td>Vector of pressures (Output)</td>
</tr>
<tr>
<td>DPDR</td>
<td>Vector of $\partial P/\partial \rho$ (Output)</td>
</tr>
<tr>
<td>DPDT</td>
<td>Vector of $\partial P/\partial T$ (Output)</td>
</tr>
<tr>
<td>E</td>
<td>Vector of specific internal energies (Output)</td>
</tr>
<tr>
<td>DEDR</td>
<td>Vector of $\partial E/\partial \rho$ (Output)</td>
</tr>
<tr>
<td>DEDT</td>
<td>Vector of $\partial E/\partial T$ (Output)</td>
</tr>
<tr>
<td>Z</td>
<td>Vector of charge states (Output)</td>
</tr>
<tr>
<td>SR</td>
<td>Vector of Rosseland opacities (Output)</td>
</tr>
<tr>
<td>SP</td>
<td>Vector of Planck opacities (Output)</td>
</tr>
<tr>
<td>XLOG</td>
<td>Scratch vector to hold $\log_{10} \rho$ (Output)</td>
</tr>
<tr>
<td>YLOG</td>
<td>Scratch vector to hold $\log_{10} T$ (Output)</td>
</tr>
<tr>
<td>KMAX</td>
<td>Number of elements in vectors (Input)</td>
</tr>
<tr>
<td>IRAD</td>
<td>Switch to determine whether opacities are computed = 1 yes, = 0 no (Input)</td>
</tr>
</tbody>
</table>
PROGRAM CREATELIB

PURPOSE: TO CONTROL THE SUBROUTINE UPDATE TO CREATE /
UPDATE A SESAME II BINARY LIBRARY

REMARKS: FIVE FILES ARE OPENED

UNIT 1 -- FILE CONTAINING SESAME DATA IN CARD IMAGE FORM
UNIT 2 -- FILE CONTAINING OLD SESAME II LIBRARY IN BINARY
UNIT 3 -- A SCRATCH FILE
UNIT 4 -- FILE CONTAINING NEW SESAME II LIBRARY IN BINARY
UNIT 5 -- FILE CONTAINING MESSAGES GENERATED DURING UPDATE

EXTERNALS: UPDATE

PROGRAMER: R. R. PETERSON

DATE: 5 MAY 1986

DATA INEW/0/
DATA DATE/050586/
DATA FMT/4H20A4/
OPEN( UNIT=1, STATUS='OLD', FILE='SESTST.DAT', READONLY )
OPEN( UNIT=2, STATUS='UNKNOWN', FILE='CURLIB.DAT',
1 FORM='UNFORMATTED' )
OPEN( UNIT=3, STATUS='SCRATCH', FILE='SESTMP.DAT',
1 FORM='UNFORMATTED' )
OPEN( UNIT=4, STATUS='NEW', FILE='NEWLIB.DAT',
1 FORM='UNFORMATTED' )
OPEN( UNIT=5, STATUS='NEW', FILE='SES.PRT', FORM='FORMATTED' )
CALL UPDATE( DATE, FMT, INEW, 1, 2, 3, 4, 5 )
STOP
END

SUBROUTINE UPDATE(DATE, NCPW, FMT, INEW, LUP, LIB, LM, LNEW, LP)

SUBROUTINE TO CREATE / UPDATE A SESAME II LIBRARY

ARGUMENTS DATE (INPUT) FLOATING PT. DATE MMDDYY
NCPW (INPUT) NO. OF HOLLERITH CHARACTERS/WORD
FMT (INPUT) FORMAT FOR READING HOLLERITH CARD
INEW (INPUT) 1=UPDATE OLD LIBRARY
0=CREATE NEW LIBRARY
LUP (INPUT) CODED UPDATE FILE UNIT NO.
LIB (INPUT) CURRENT LIBRARY UNIT NO.
LM (INPUT) UNIT NO. OF SCRATCH FILE
LNEW (INPUT) UNIT NO. OF CREATED/UPDATED LIBRARY
LP (INPUT) PRINTER UNIT NO.

REMARKS NONE
EXTERNALS S2BLDM, S2SRTM, S2BOLDL, S2UPDT, COPMTS

PROGRAMMER: J. ABDALLAH, JR.

DATE: 20 FEB 1979

DIMENSION RMATS(200), INDEX(200)
WRITE(LP, 1000)

BUILD MATERIAL FILE FROM UPDATE FILE
CALL S2BLDM(LUP, LM, LNEW, LP, LNEW, DATE, NCPW, FMT, NMATS, RMATS, INDEX)

IF NO MATERIALS FOUND, GO BACK
IF(NMATS.LE.0) GO TO 100

SORT MATERIAL FILE ACCORDING TO MATERIAL AND TABLE ID
WRITE(LP, 1010)
CALL S2SRTM(LM, LNEW, NMATS, RMATS, INDEX)
IF(INEW.EQ.1) GO TO 10

BUILD NEW LIBRARY HERE
CALL COPMTS(LNEW, LM, NMATS, INDEX)
CALL S2BOLD(LM, LNEW, DATE, 1.0)
WRITE(LP, 1050)
RETURN

UPDATE EXISTING LIBRARY HERE
WRITE(LP, 1030)
CALL S2UPDT(LNEW, LIB, LM, DATE, NMATS, RMATS, INDEX)
WRITE(LP, 1060)
RETURN

RETURN

1000 FORMAT(1X, 39H CREATING MATERIAL FILE FROM UPDATE FILE)
1010 FORMAT(1X, 21HSORTING MATERIAL FILE)
1020 FORMAT(1X, 20H CREATING NEW LIBRARY)
1030 FORMAT(1X, 25H UPDATING EXISTING LIBRARY)
1040 FORMAT(1X, 31H ERROR IN PROCESSING UPDATE FILE)
1050 FORMAT(1X, 17H CREATION COMPLETE)
1060 FORMAT(1X, 15H UPDATE COMPLETE)
END

SUBROUTINE S2BLDM(LUP, LM, LSCR, LP, LNEW, DATE, NCPW, FMT, NMATS, RMATS, INDEX)

PURPOSE TO BUILD A MATERIAL FILE FROM A CODED UPDATE FILE

ARGUMENTS LUP (INPUT) UPDATE FILE UNIT NO.
LM (INPUT) OUTPUT MATERIAL FILE UNIT NO.
LSCR (INPUT) SCRATCH FILE UNIT NO.
LP (INPUT) PRINTER UNIT NO.
LNEW (INPUT) 0=CREATE NEW LIBRARY
1=UPDATE EXISTING LIBRARY
DATE (INPUT) FLOATING PT. DATE
NCPW (INPUT) NO OF CHARACTERS PER WORD
FMT (INPUT) FORMAT FOR READING CARD IMAGE
NMATS (OUTPUT) NUMBER OF MATERIALS OBTAINED ON LM
RMATS (OUTPUT) ARRAY OF MATERIAL IDS
INDEX (OUTPUT) ARRAY NO OF WORDS IN MATERIAL INDICES
REMARKS  A MATERIAL FILE IS A FILE IN SESAME FORMAT EXCEPT
       THAT THE FILE DIRECTORY HAS BEEN EXCLUDED

EXTERNALS  RETMAT, BDINDX, INBUF, OUTBUF, UPSUM

PROGRAMMER  J. ABDALLAH, JR.

DATE  20 FEB 1979

DIMENSION RMATS(1), INDEX(1), HINDEX(35), NWTAB(15), TBLIDS(15)
       COMMON/SES.COM/BUFF(10000)
       REWIND LM
       REWIND LUP

10  CALL RETMAT(LUP, LSCR, NCPW, FMT, RMAT, DATE1, DATE2, VERS,
       NTAB, TBLIDS, NWTAB, IFLAG)
       IF(IFLAG.NE.0) GO TO 1000
       IF(DATE1.NE.0.0) GO TO 20
       DATE1=DATE
       DATE2=DATE
       GO TO 30
20  IF(DATE2.EQ.0.0) DATE2=DATE
30  IF(INEW.EQ.0 AND VERS.EQ.0.0) VERS=1.0
       I=1
       RMATS(I)=RMAT
       CALL BDINDX(RMAT, DATE1, DATE2, VERS, NTAB, TBLIDS, NWTAB, HINDEX)
       CALL UPSUM(LP, I, RMAT, DATE1, NTAB, TBLIDS, NWTAB)
       NWI=NTAB+NTAB+5
       INDEX(I)=NWI
       CALL OUTBUF(LM, HINDEX, NWI)
       REWIND LSCR
       DO 100 J=1, NTAB
       NW=NTAB(J)
       IF(NW.EQ.0) GO TO 100
       CALL INBUF(LSCR, BUFF, NW, IM)
       CALL OUTBUF(LM, BUFF, NW)
100  CONTINUE
       CALL OUTBUF(LM, BUFF, 0)
       GO TO 10
1000  RMATS=I
       CALL OUTBUF(LM, BUFF, 0)
       RETURN
       END

SUBROUTINE RETMAT(LUP, LS, NCPW, FMT, RMAT, DATE1, DATE2, VERS,
       NTAB, TBLIDS, NWTAB, IFLAG)

                  SUBROUTINE  RETMAT(LUP, LS, NCPW, FMT, RMAT, DATE1, DATE2, VERS,
                  NTAB, TBLIDS, NWTAB, IFLAG)
                  PURPOSE  TO RETRIEVE THE DATA FOR ONE MATERIAL FROM THE
                  UPDATE FILE
                  ARGUMENTS  LUP  (INPUT)  UPDATE FILE UNIT NO.
                  LS  (INPUT)  UNIT TO WHICH TABLES ARE WRITTEN
                  NCPW  (INPUT)  NO. OF CHARMS. / WORD
                  FMT  (INPUT)  FORMAT FOR READING CARD IMAGE
                  DATE1  (OUTPUT)  CREATION DATE
C       DATE2  (OUTPUT)  DATE OF LAST UPDATE
C       VERS   (OUTPUT)  VERSION NO.
C       RMAT   (OUTPUT)  MATERIAL ID
C       NTAB   OUTPUT)  NO. OF TABLES
C       TBLIDS (OUTPUT)  ARRAY OF TABLE IDS
C       NWTAB  (OUTPUT)  NO. OF WORDS PER TABLE
C       IFLAG  (OUTPUT)  1=END OF FILE

REMARKS  DATA TABLES ARE WRITTEN ON LS IN THE
          ORDER THEY ARE READ OFF THE UPDATE FILE

EXTERNALS RDCARD, OUTBUF
C
C PROGRAMMER  J. ABDALLAH, JR.
C
C DATE       20 FEB 1979
C
C------------------------------------------------------
DIMENSION TBLIDS(1), NWTAB(1)
COMMON/SESOM/BUFF(10000)
DATA IDEL/10/4/>
DATA JFLAG/0/
IFLAG=0
NTAB=0
C . READ MARKER CARD
REWIND LS
IF (JFLAG, NE, 0) GO TO 110
JFLAG = 1
10 READ(LUP, 100) IFM, RATID, TBLID, NWDC, IDIS, DATEA, DATEB, UVERS
100 FORMAT(12, 2F6.0, 16, 3X, A1, 2F9.0, F4.0)
110 IF (NTAB, EQ, 0, AND, IFLAG, EQ, 2) GO TO 2000
    IF (IFM, EQ, 2) GO TO 2010
    IF (NTAB, NE, 0, AND, IFLAG, EQ, 0) GO TO 2010
    DATE1 = DATEA
    DATE2 = DATEB
    VERS = UVERS
    NTAB = NTAB + 1
    TBLIDS(NTAB) = TBLID
    IF (TBLID, LT, 200.0) NWDC = (NWDC + NCPW - 1) / NCPW
    NWTAB(NTAB) = NWDC
    NWPC = 80
    IF (NWPC, NCPW, LT, 80) NWPC = NWPC + 1
    RMAT = RATID
    IF (IDIS, EQ, IDEL) GO TO 150
    CALL RDCARD(LUP, BUFF, TBLID, NWDC, NWPC, FMT)
    CALL OUTBUF(LS, BUFF, NWDC)
    GO TO 10
150 NWTAB(NTAB) = 0
    GO TO 10
2000 IFLAG = 1
2010 RETURN
END
C------------------------------------------------------

SUBROUTINE BDINDEX(RMAT, DATE1, DATE2, VERS, NTAB, TBLIDS, NWTAB, MINDEX)
C
C SUBROUTINE  BDINDEX(RMAT, DATE1, DATE2, VERS, NTAB, TBLIDS, NWTAB, MINDEX)
C
C PURPOSE  TO BUILD THE MATERIAL INDEX
C
C ARGUMENTS  RMAT  (INPUT)  MATERIAL I.D.
C DATE1  (INPUT)  CREATION DATE
C DATE2  (INPUT)  DATE OF LAST UPDATE
C VERS  (INPUT)  VERSION NUMBER
C NTAB  (INPUT)  NO. OF TABLES
C TBLIDS  (INPUT)  ARRAY OF TABLE IDS
C NWTAB  (INPUT)  ARRAY OF NO. OF WORDS/TABLE
C HINDEX  (OUTPUT)  FLOATING POINT MATERIAL INDEX
C
C REMARKS  NONE
C C EXTERNALS  NONE
C C PROGRAMMER  J.ABDALLAH,JR.
C C DATE  20 FEB 1979
C
C---------------------------------------------------------------
C
C DIMENSION TBLIDS(1),NWTAB(1),HINDEX(1)
C HINDEX(1)=RMAT
C HINDEX(2)=DATE1
C HINDEX(3)=DATE2
C HINDEX(4)=VERS
C HINDEX(5)=FLOAT(NTAB)
C DO 10 J=1,NTAB
C HINDEX(5+J)=TBLIDS(J)
C HINDEX(5+J+NTAB)=FLOAT(NWTAB(J))
C 10 CONTINUE
C RETURN
C END
C
C SUBROUTINE S2SRTM(LM,LMS,N,RMATS,INDEX)
C---------------------------------------------------------------
C
C SUBROUTINE S2SRTM(LM,LMS,N,RMATS,INDEX)
C PURPOSE  TO SORT MATERIAL FILE IN INCREASING MATERIAL AND
C TABLE IDS
C C ARGUMENTS  LM  (INPUT)  UNIT NO. OF INPUT MATERIAL FILE
C LMS  (INPUT)  UNIT NO. OF OUTPUT SORTED FILE
C N  (INPUT)  NO. OF MATERIAL IDS
C RMATS  (INPUT)  ARRAY OF MATERIAL IDS
C INDEX  (INPUT)  NO. OF WORDS IN EACH MATERIAL INDEX
C C REMARKS  THE CALL TO S2SRTM MAY BE SKIPPED IF FILE IS ALREADY
C SORTED
C C EXTERNALS  ASORT2,INBUF,OUTBUF,TFCH,COMPACK
C C PROGRAMMER  J.ABDALLAH,JR.
C C DATE  20 FEB 1979
C
C---------------------------------------------------------------
C
C DIMENSION RMATS(1),INDEX(1)
C COMMON/SESC/BUF(10000)
C REWIND LM
C REWIND LMS
C DO 1 J=1,N
C BUFF(J)=INDEX(J)
1 CONTINUE
   CALL ASORT2(N,RMATS,BUFF(1))
   IPT1=N+1
   NX=0
   DO 10 J=1,N
      RM=RMATS(J)
   10 IF(NX.NE.0) GO TO 12
      REWIND LM
      GO TO 14
   12 CALL SKIPMF(LM,1)
   NX=NX+1
   NWDS=INDEX(NX)
   CALL INBUF(LM,BUFF(IPT1),NWDS,IFLG)
   IF(NX.NE.N) GO TO 15
   NX=0
   15 IF(BUFF(IPT1).EQ.RM) GO TO 20
      GO TO 10
   20 NT=BUFF(IPT1+4)
      IPT2=IPT1+NWDS
      IF(NT.EQ.1) GO TO 50
      T1=BUFF(IPT1+5)
      DO 30 K=2,NT
         IF(BUFF(IPT1+4+K).LT.T1) GO TO 100
         T1=BUFF(IPT1+4+K)
   30 CONTINUE
   C . . NO SORT ON TABLES REQUIRED - COPY THEM
   50 CALL OUTBUF(LMS,BUFF(IPT1),NWDS)
      DO 80 K=1,NT
         NTW=BUFF(IPT1+4+K+NT )
      80 IF(NTW.EQ.0) GO TO 80
         CALL INBUF(LM,BUFF(IPT2),NTW,IFLG)
      CALL OUTBUF(LMS,BUFF(IPT2),NTW)
   80 CONTINUE
   GO TO 900
   C . . TABLE SORT REQUIRED
   100 CALL COMPAK(BUFF(IPT1),BUFF(IPT2))
      NT3=BUFF(IPT2+4)
      IPT3=IPT2+NT3+NT3+5
      CALL ASORT2(NT,BUFF(IPT1+5),BUFF(IPT1+5+NT))
      CALL OUTBUF(LMS,BUFF(IPT1),NWDS)
      BACKSPACE LM
      DO 130 K=1,NT
         K5=IPT1+4+K
      130 IF(BUFF(K5+NT).EQ.0.0) GO TO 130
         CALL TBFCH(BUFF(K5),BUFF(IPT3),BUFF(IPT2),LM,9765,IFL)
      CALL OUTBUF(LMS,BUFF(IPT2),IFL)
   100 CONTINUE
   900 CALL OUTBUF(LMS,BUFF,0)
   1000 CONTINUE
      CALL OUTBUF(LMS,BUFF,0)
      DO 1010 J=1,N
         INDEX(J)=BUFF(J)
   1010 CONTINUE
   RETURN
END

SUBROUTINE S2BLDL(LM,LNEW,DATE,VERS)

C-----------------------------------------------------------------------
C SUBROUTINE S2BLDL(LM,LNEW,DATE,VERS)
C PURPOSE TO BUILD A SESAME II LIBRARY FROM A MATERIAL FILE
C
C ARGUMENTS
C LM (INPUT) MATERIAL FILE UNIT NO.
C LNEW (INPUT) LIBRARY UNIT NO.
C DATE (INPUT) FLOATING POINT DATE
C VERS (INPUT) VERSION NUMBER
C
C REMARKS SIMBDL BUILDS THE MASTER DIRECTORY FROM THE MATERIAL
C FILE AND COPIES IT AND THE DATA FROM THE
C MATERIAL FILE TO THE LIBRARY UNIT TO CREATE A
C SESAME II TYPE FILE
C
C EXTERNALS SKIPMF,INBUF,COPMAT
C
C PROGRAMMER J.ABDALLAH,JR.
C
C DATE 20 FEB 1979
C
CCOMMOM/SES.COM/BUFF(6000),IBUFF(3000),BUFI(1000)
C IA=1
C IM=3
C IN=3000
C IA=0
C IPTM=IM
C IPTN=IN
C IFTA=IA
C REWIND LM
C
C BUILD TABLES FOR DIRECTORY
C NM=0
C GO TO 2
C 1 CALL SKIPMF(LM,1)
C 2 CALL INBUF(LM,BUFF,5,IFLG)
C IP(IFLG.EQ.0) GO TO 1000
C NM=NM+1
C IPTM=IPTM+1
C IPTN=PTN+1
C IFTA=IFTA+1
C IBUFF(IFTA)=IA
C BUFI(IPTN)=BUFI(1)
C NT=BUFI(5)
C NINDX=5+NT+NT
C BACKSPACE LM
C CALL INBUF(LM,BUFF,NINDX,IFLG)
C BUFI(IPTN)=FLOAT(NINDX)
C IAO=IAO+NINDX+1
C DO 10 J=1,NT
C NW=BUFI(5+NT+J)
C IAO=IAO+NW+1
C 10 CONTINUE
C IAO=IAO+1
C GO TO 1
C
C MOVE DIRECTORY TO PROPER PLACES
C 1000 BUFI(1)=FLOAT(NM)
C BUFI(2)=DATE
C BUFI(3)=VERS
C REWIND LNEW
C CALL OUTBUF(LNEW,BUFF,3)
C NM3=3*N
C NDIR=4+NM3+2
C . . BUILD SECOND RECORD OF DIRECTORY
C . . MOVE MATERIAL NO. TABLE
IPT=0
DO 1100 J=1,NM
IPT=IPT+1
IM=IM+1
BUFF(IPT)=BUFF(IM)
1100 CONTINUE
C . . MOVE INDEX TABLE
IPTN=IN
DO 1200 J=1,NM
IPT=IPT+1
IPTN=IPTN+1
BUFF(IPT)=BUFF(IPTN)
1200 CONTINUE
C . . MOVE ADDRESS TABLE
DO 1300 J=1,NM
IPT=IPT+1
IA=IA+1
IAD=IAD+1
1300 CONTINUE
CALL OUTBUF(LNEW,BUFF,NM3)
CALL OUTBUF(LNEW,BUFF,0)
C . . COPY DATA TABLES FROM LM TO LNEW
REWIND LM
IPTN=IN
DO 1400 J=1,NM
IPTN=IPTN+1
1400 BUFF(J)=BUFF(IPTN)
IPT=NM+1
DO 3000 J=1,NM
IF(J.NE.1) CALL SKIPFM(LM,J)
BUFF(J)=BUFF(J)
CALL COPMAT(LM,LNEW,BUFF(IPT),NW)
3000 CONTINUE
CALL OUTBUF(LNEW,BUFF,0)
RETURN
END

SUBROUTINE S2UPDT(LM,LIB,LSCR,DATE,NM,RMATS,INDEX)

SUBROUTINE S2UPDT(LM,LIB,LSCR,DATE,NM,RMATS,INDEX)

PURPOSE TO UPDATE THE CURRENT SESAME LIBRARY AND BUILD
A NEW SESAME LIBRARY

ARGUMENTS LM (INPUT) UNIT NO. OF MATERIAL FILE CONTAINING
UPDATES - IT IS REPLACED BY THE
SESAME II LIBRARY UPON EXIT
LIB (INPUT) UNIT NO. OF CURRENT LIBRARY
LSCR (INPUT) UNIT NO. OF SCRATCH FILE
NM (INPUT) NO OF MATERIALS ON MATERIAL FILE
RMATS (INPUT) ARRAY OF MATERIAL IDS ON MATERIAL
FILE
INDEX (INPUT) NO. OF WORDS IN EACH MATERIAL INDEX
ON MATERIAL FILE
DATE (INPUT) FLOATING POINT DATE
C REMARKS NONE
C EXTERNALS INBUF,OUMBUF,UPTPGM,SKIPMF,S2MOD,COPMAT,S2BLDL
C PROGRAMMER J.ABDALLAH,JR.
C DATE 20 FEB 1979
C
C DIMENSION RMATS(1),INDEX(1),IUPDT(200)
COMMON/SESOCOM/BUFF(10000)
C
C GET SESAME DIRECTORY
C REWIND LIB
C REWIND LM
C REWIND LSCR
C CALL INBUF(LIB,BUFF,3,IFLG)
C IM=0
C N=BUFF(1)
C VERS=BUFF(3)
C IL=0
C N= N+N
C IPTS=N+1
C NSP=10000-IPTS+1
C CALL INBUF(LIB,BUFF(4),N3,IFLG)
C CALL UPTPGN(N,BUFF(4),NM,ROATS,NUT,IUPDT)
C DO 300 J=1,N
C BUFF(J)=BUFF(3+N+J)
C 300 CONTINUE
C DO 1000 J=1,NUT
C IDJ=IUPDT(J)
C IF(IDJ) 600,400,500
C 400 IL=IL+1
C IM=IM+1
C NWI=BUFF(IL)
C IF(IM.GT.1) CALL SKIPMF(LM,1)
C CALL SKIPMF(LIB,1)
C CALL S2MOD(LM,LIB,LSCR,BUFF(IPTS),NSP,INDEX(IM),NWI)
C GO TO 1000
C 500 IM=IM+1
C IF(IM.GT.1) CALL SKIPMF(LM,1)
C CALL COPMAT(LM,LSCR,BUFF(IPTS),INDEX(IM))
C GO TO 1000
C 600 IL=IL+1
C CALL SKIPMF(LIB,1)
C NWI=BUFF(IL)
C CALL COPMAT(LIB,LSCR,BUFF(IPTS),NWI)
C 1000 CONTINUE
C CALL OUTBUF(LSCR,BUFF,0)
C VERS=VERS+1.0
C CALL S2BLDL(LSCR,LM,DAT,VRS)
C RETURN
C END
C
C SUBROUTINE COPMAT(L1,L2,STORE,NWI)
C
C SUBROUTINE COPMAT(L1,L2,STORE,NWI)
C
C PURPOSE TO COPY A MATERIAL FROM ONE FILE TO ANOTHER
C
C ARGUMENTS L1 (INPUT) COPY FROM UNIT NO.
C L2 (INPUT) COPY TO UNIT NO.
C STORE (INPUT) STORAGE AREA FOR DATA RECORDS
C NWI (INPUT) NO. OF WORDS IN MATERIAL INDEX
C IF NWI.LT.0 THE NUMBER OF WORDS IS COMPUTED
C
C REMARKS L1 AND L2 MUST POINT TO THE INDEX RECORD
C EXTERNALS INBUF,OUTBUF
C PROGRAMMER J.ABDALLAH,JR.
C DATE 20 FEB 1979
C
C-----------------------------------------------
C
C DIMENSION STORE(1)
C NW=5
C IF(NWI.GT.0) NW=NWI
C CALL INBUF(L1,STORE,NW,IFLG)
C NT=STORE(5)
C IF(NWI.GT.0) GO TO 10
C NW=5+NT+NT
C BACKSPACE L1
C CALL INBUF(L1,STORE,NW,IFLG)
C 10 IPT=NW+1
C CALL OUTBUF(L2,STORE,NW)
C WRITE REMAINING DATA TABLES
C DO 50 J=1,NT
C NW=STORE(5+J+NT)
C CALL INBUF(L1,STORE(IPT),NW,IFLG)
C CALL OUTBUF(L2,STORE(IPT),NW)
C 50 CONTINUE
C CALL OUTBUF(L2,STORE,0)
C RETURN
C END
C
C SUBROUTINE S2MOD(LM,LIB,LMU,STORE,LENS,NWM,NWL)
C-----------------------------------------------
C
C SUBROUTINE S2MOD(LM,LIB,LMU,STORE,LENS,NWM,NWL)
C PURPOSE TO MODIFY THE LIBRARY MATERIAL FILE USING THE
C BINARY UPDATE FILE
C ARGUMENTS LM (INPUT) UNIT NO. OF UPDATE FILE
C LIB (INPUT) LIBRARY VERSION OF MATERIAL UNIT NO.
C LMU (INPUT) UNIT NO OF UPDATED MATERIAL
C STORE (INPUT) STORAGE AREA
C LENS (INPUT) LENGTH OF STORAGE AREA
C NWM (INPUT) NO. OF WORDS IN MAT. INDEX ON UPDATE FILE
C NWL (INPUT) NO. OF WORDS IN INDEX OF LIBRARY VERSION
C
C REMARKS LM AND LIB MUST POINT TO THE INDEX RECORD
C EXTERNALS COMBIN,COMPAK,INBUF,OUTBUF,TBFCH
C PROGRAMMER J.ABDALLAH,JR.
C DATE 20 FEB 1979
DIMENSION STORE(1)
IM=1
IM=1+NWM
IL=IL+NWM
IN=IN1+NWL+NWM
ISTX=IN1+NWL+NWM
LNX=LENS-ISTX+1
CALL INBUF(LM,STORE(IM1),NWM,IFLG)
CALL INBUF(LIB,STORE(IL),NWL,IFLG)
CALL COMBIN(STORE(IM1),STORE(IL),STORE(IN1))
CALL COMP(STORE(IM1),STORE(IM))
CALL COMP(STORE(IN1),STORE(IN))

C . CHECK FOR TOTAL DELETED
NT=STORE(IN+4)
IF(NT.EQ.0) RETURN
NWDS=NT+NT+5
CALL OUTBUF(LMU,STORE(IN),NWDS)
BACKSPACE LM
BACKSPACE LIB
DO 100 K=1,NT
TID=STORE(IN+4+K)
IF(STORE(IM+4).EQ.0.0) GO TO 60
C . SEARCH UPDATE FILE FOR TABL
CALL TBCH(TID,STORE(ISTX),STORE(IM),LM,LNX,NUM)
IF(NUM.GT.0) GO TO 80
C . SEARCH LIB FILE FOR TABLE
60 CALL TBCH(TID,STORE(ISTX),STORE(IL),LIB,LNX,NUM)
80 CALL OUTBUF(LMU,STORE(ISTX),NUM)
100 CONTINUE
CALL OUTBUF(LMU,STORE,0)
RETURN

SUBROUTINE COMP(IMA,IMB)

SUBROUTINE COMP(IMA,IMB)

PURPOSE TO COMPACT MATERIAL INDEX BY REMOVING
TABLE DELETION INDICATORS

ARGUMENTS IMA (INPUT) INDEX TO BE COMPACTED
IMB (OUTPUT) COMPACTED INDEX

REMARKS NONE

EXTERNALS NONE

PROGRAMMER J. ABDALLAH

DATE 20 FEB 1979

COPY FIRST FOUR WORDS OF INDEX
DEXB(1)=DEXA(1)
DEXB(2)=DEXA(2)
DEXB(3)=DEXA(3)
DEXB(4)=DEXA(4)
C. COMPUTE THE NUMBER OF TABLES IN THE COMPACTED INDEX
   NA=DEXA(5)
   NB=0
   DO 10 J=1,NA
       IF(DEXA(J+NA+5).EQ.0.0) GO TO 10
       NB=NB+1
   10 CONTINUE
       DEXB(5)=FLOAT(NB)
       IF(NB.EQ.0.0) RETURN
   C. MOVE TABLE IDS INTO INDEX
   JS=5
   KS=5
   IPTA=5+NA
   IPT=5+NB
   I=0
   DO 30 J=1,NA
       IPTA=IPTA+1
       JS=JS+1
       IF(DEXA(IPTA).EQ.0.0) GO TO 30
       I=I+1
       IPT=IPT+1
       KS=KS+1
       DEXB(IPT)=DEXA(IPTA)
       DEXB(KS)=DEXA(JS)
   30 CONTINUE
   40 CONTINUE
   RETURN
   SUBROUTINE UPTPGM(NL,RMATL,NM,RMATM,NUT,IPGM)

   C SUBROUTINE UPTPGM(NL,RMATL,NM,RMATM,NUT,IPGM)
   PURPOSE UPTPGM(NL,RMATL,NM,RMATM,NUT,IPGM)
   ARGUMENTS
   NL (INPUT) NO. OF MATERIALS ON LIBRARY
   RMATL (INPUT) MATERIAL IDS ON LIBRARY
   NM (INPUT) NO. OF MATERIALS ON UPDATE FILE
   RMATM (INPUT) ARRAY OF MATERIALS ON UPDATE FILE
   NUT (INPUT) NO. OF MATERIALS IN UPDATED LIBRARY
   IPGM (OUTPUT) PROGRAM ARRAY
       -1=USE OLD MATERIAL AS IS
       0=UPDATE MATERIAL
       1=USE NEW MATERIAL AS IS
   REMARKS THE IPGM ARRAY IS USED TO CONTROL PROCESSING IN S2UPDT
   EXTERNALS NONE
   PROGRAMMER J.ABDALLAH
   DATE 20 FEB 1979

   DIMENSION RMATL(1),RMATM(1),IPGM(1)
   IL=1
IM=1
NUT=0
10 RSM=RMATM(IM)
   RSL=RMATL(IL)
   NUT=NUT+1
   IF(RSL=RSM) 100,200,300
100 IL=IL+1
   INUM(NUT)=-1
   GO TO 500
200 INUM(NUT)=0
   IL=IL+1
   IM=IM+1
   GO TO 500
300 INUM(NUT)=1
   IM=IM+1
500 IF(IM.LE.NM.AND.II.LE.NL) GO TO 10
   IF(IM.GT.NM.AND.II.GT.NL) GO TO 600
   NUT=NUT+1
   IF(IM.GT.NM) GO TO 100
   IF(IIL.GT.NL) GO TO 300
600 CONTINUE
RETURN
END

SUBROUTINE COMBIN(DEXM,DEXL,DEXN)

C------------------------------------------------------------------------
SUBROUTINE COMBIN(DEXM,DEXL,DEXN)

C PURPOSE TO COMBINE THE MATERIAL INDEX OF THE UPDATE FILE WITH THE INDEX
OF THE LIBRARY VERSION TO FORM A NEW MATERIAL INDEX

C ARGUMENTS

DEXM (INPUT) INDEX OF MATERIAL ON UPDATE FILE
DEXL (INPUT) INDEX OF MATERIAL ON LIBRARY
DEXN (OUTPUT) NEW UPDATED INDEX

C REMARKS THE RESULTING INDEX INCLUDES DELETION INDICATORS
(IF ANY)

C EXTERNALS NONE

C PROGRAMMER J.ABDALLAH,JR.

C DATE 20 FEB 1979

C------------------------------------------------------------------------

DIMENSION DEXM(1),DEXL(1),DEXN(1)
C . . COMPUTE THE TOTAL NUMBER OF TABLES
C . . IN UPDATED MATERIAL
NTM=DEXM(5)
NTL=DEXL(5)
NTOT=NTL
DO 20 K=1,NTM
   TM=DEXM(5+K)
   DO 10 J=1,NTL
   IF(TM.EQ.DEXL(J+5)) GO TO 20
10 CONTINUE
   NTOT=NTOT+1
20 CONTINUE
C . . BUILD TABLE ARRAY
IM=1
IL=1
DO 500 J=1,NTOT
TM=DEXM(5+IM)
NWM=DEXM(5+IM+NTM)
TL=DEXL(5+IL)
NWL=DEXL(5+IL+NTL)
IF(IM.GT.NTM) GO TO 180
IF(IL.GT.NTL)  GO TO 140
IF(TM-TL)  140,160,180
140 IM=IM+1
DEXN(5+J)=TM
DEXN(5+J+NTOT)=FLOAT(NWM)
GO TO 500
160 IM=IM+1
IL=IL+1
DEXN(J+5)=TM
DEXN(J+5+NTOT)=FLOAT(NWM)
GO TO 500
180 IL=IL+1
DEXN(5+J)=TL
DEXN(5+J+NTOT)=FLOAT(NWL)
500 CONTINUE
C . MOVE THE REMAINDER OF THE NEW INDEX TO APPROPRIATE SPOTS
DEXN(1)=DEXL(1)
DEXN(2)=DEXL(2)
DEXN(3)=DEXM(3)
VERS=DEXM(4)
IF(VERS.EQ.0.0) VERS=DEXL(4)+1.0
DEXN(4)=VERS
DEXN(5)=FLOAT(NTOT)
RETURN
END

SUBROUTINE ASORT2(N,A,B)

DIMENSION A(1),B(1)
IF(N.EQ.1) RETURN
I=1
50 I=I+1
DO 100 J=11,N
IF(A(J).GT.A(I)) GO TO 100
ASAVE=A(I)
A(I)=A(J)
100 CONTINUE

A(J)=ASAVE
BSAVE=B(I)
B(I)=B(J)
B(J)=BSAVE

100 CONTINUE
   I=I+1
   IF(I.LT.N) GO TO 10
   RETURN
END

SUBROUTINE UPSUM(LP, I, RMAT, DATE, NTAB, TBLIDS, NWTAB)

C-------------------------------------------------------
C
C SUBROUTINE UPSUM(LP, I, RMAT, DATE, NTAB, TBLIDS, NWTAB)
C
C PURPOSE: TO PRINT THE UPDATE SUMMARY
C
C ARGUMENTS
   LP (INPUT) PRINTER UNIT NO.
   I (INPUT) RELATIVE MAT NO. IN UPDATE FILE
   RMAT (INPUT) MATERIAL ID
   NTAB (INPUT) NO. OF TABLES
   TBLIDS (INPUT) ARRAY OF TABLE IDS
   NWTAB (INPUT) NO. OF WORDS IN EACH TABLE

C
C REMARKS
   NONE

C EXTERNALS
   NONE

C PROGRAMMER
   J. ABDALLAH

C DATE
   27 FEB 1979

C-------------------------------------------------------

DIMENSION TBLIDS(1), NWTAB(1)
IF(I.GT.1) GO TO 10
WRITE(LP, 1000)

1000 FORMAT(1H1, 18X, 24HSESAME II UPDATE SUMMARY, //,
\$3X, 11H MATERIAL ID, 5X, BMCREATION, 7X, BHTABLE ID,
\$5X, 2XHNO. OF WORDS(O=DELETION))

10 MATID=RMAT
   IDATE=DATE
   DO 40 J=1, NTAB
      ITBL=TBLIDS(J)
      NWDS=NWTAB(J)
      IF(J.GT.1) GO TO 30
      WRITE(LP, 1010) I, MATID, IDATE, ITBL, NWDS
   GO TO 40
1010 WRITE(LP, 1020) ITBL, NWDS

40 CONTINUE

1020 FORMAT(1X, 12, 2X, 16, 9X, 16, 7X, 16, 9X, 16)

RETURN
END

SUBROUTINE OUTBUF(LU, Z, N)

C-------------------------------------------------------
C
C SUBROUTINE: OUTBUF(LU, Z, N)
C
C PURPOSE: WRITE DATA OR END FILE MARK ON DISC OR TAPE

C-------------------------------------------------------
C
C ARGUMENTS:  LU (INPUT) - NUMBER OF TAPE, DISC, ETC.
Z (INPUT) - NAME OF LOCAL ARRAY
N (INPUT) - NUMBER OF WORDS TO BE WRITTEN
         FOR N.LE.0, WRITES AN END FILE MARK
C
C REMARKS: THIS ROUTINE IS JUST AN BINARY WRITE OR
END FILE MARK. THE USER MAY WISH TO MODIFY IT
AND MAKE CORRESPONDING CHANGES TO INBUF.
C
C REMARKS:  SYSTEM BINARY WRITE AND END FILE.
C
C
C DATE:  7 DECEMBER 1976
C
C-------------------------------------------------------------------
C****  LCM Z
DIMENSION Z(1)
IF(N.GT.0) GO TO 1
ENDFILE LU
RETURN
1 WRITE(LU)(Z(I),I=1,N)
RETURN
END
SUBROUTINE INBUF(LU,Z,N,IM)
C
C SUBROUTINE:  INBUF(LU,Z,N,IM)
C PURPOSE:  READ FROM DISC OR TAPE INTO LOCAL ARRAY
C
C ARGUMENTS:  LU (INPUT) - NUMBER OF TAPE, DISC, ETC.
Z (IN/OUT) - NAME OF LOCAL ARRAY
N (INPUT) - NUMBER OF WORDS TO BE READ
IM (OUTPUT) - END FILE FLAG
         IM = 0, IF END FILE MARK ENCOUNTERED
         IM = 1, IF NO END FILE MARK
C
C REMARKS:  THIS ROUTINE IS JUST AN BINARY READ AND TEST
FOR END OF FILE. IT ASSUMES THE DATA RECORDS
WERE WRITTEN WITH OUTBUF AND SHOULD BE MADE
COMPATIBLE WITH THAT ROUTINE. NOTE THAT THE
TWO BRANCH END FILE TEST IS NOT STANDARD.
C
C EXTERNALS:  SYSTEM BINARY READ, END FILE TEST.
C DATE:  6 DECEMBER 1976
C
C-------------------------------------------------------------------
C****  LCM Z
DIMENSION Z(1)
READ(LU,IOSTAT=IEOF)(Z(I),I=1,N)
IF(IEOF .GT. -1) GOTO 2
IM = 0
RETURN
2 IM = 1
RETURN
END
SUBROUTINE SKIPMF (LU,NF)

PURPOSE: TO SKIP OVER -NF- MATERIAL FILES.

ARGUMENTS: 
    LU (IN) - LOGICAL UNIT NUMBER
    NF (IN) - NUMBER OF FILES TO BE SKIPPED OVER

REMARKS: FIRST CHECKS FOR END-OF-INFORMATION, THEN READS
    ONE WORD FROM EACH SUCCEEDING RECORD UNTIL
    END-OF-FILE IS READ. LOOPS UNTIL IT HAS DONE
    THIS -NF- TIMES, OR READ EOI, WHICHEVER OCCURS
    FIRST. IF INITIALLY POSITIONED IMMEDIATELY BEFORE
    AN EOI, THEN SKIPMF SIMPLY SKIPS ONE FILE MARK AND
    RETURNS INDEPENDENT OF THE VALUE OF NF.

EXTERNALS: INBUF

PROGRAMMER: G. ROOD, T-4

DATE: 21 JAN 1977

DO 20 I=1,NF
    CALL INBUF(LU,Z,1,IEOF)
    IF(IEOF.EQ.0) GO TO 30
    10 CALL INBUF(LU,Z,1,IEOF)
    IF(IEOF.NE.0) GO TO 10
    20 CONTINUE
30 RETURN

END

SUBROUTINE COPMTS(L1,L2,N,INDEX)

PURPOSE TO COPY A MATERIAL FILE TO ANOTHER UNIT

ARGUMENTS 
    L1 (INPUT) COPY FROM UNIT NO.
    L2 (INPUT) COPY TO UNIT NO.
    N (INPUT) NO. OF MATERIAL TO BE COPIED
    INDEX (INPUT) NO. OF WORDS IN INDEX RECORDS

REMARKS NONE

EXTERNALS COPMAT

PROGRAMMER J.ABDALLAH, JR.

DATE 26 FEB 1978

DIMENSION INDEX(1)
COMMON/SESCOM/BUFF(10000)
REWRITE L1
REWRITE L2
DO 100 J=1,N
    IF(J.NE.1) CALL SKIPMF(L1,1)
100 CONTINUE
CALL COPMAT(L1,L2,BUFF,INDEX(J))
100 CONTINUE
CALL OUTBUF(L2,BUFF,0)
RETURN
END

SUBROUTINE TBFC(F,CTNR,Z,XTB,LU,NMAX,NUM)
C
SUBROUTINE: TBFC(CTNR,Z,XTB,LU,NMAX,NUM)
C
PURPOSE: GET A DATA TABLE FROM SESAME 2 LIBRARY.
C
ARGUMENTS: CTNR (INPUT) - CATALOGUE NUMBER OF DATA TABLE
Z (IN/OUT) - ARRAY WHERE TABLE IS TO BE STORED
XTB (INPUT) - MATERIAL INDEX
LU (INPUT) - LOGICAL UNIT NUMBER FOR LIBRARY
NMAX (INPUT) - MAXIMUM LENGTH OF ARRAY Z
NUM (OUTPUT) - NUMBER OF WORDS LOADED INTO Z
IF NUM = 0, IF THE TABLE CANNOT BE FOUND
NMAX IS TOO SMALL, NUM = -NUMBER OF EXTRA
WORDS NEEDED
C
REM: Z AND XTB CAN BE DECLARED LCM ON THE CDC 7600.
BEFORE CALLING, THE POINTER ON UNIT LU MUST BE
SET TO THE FIRST WORD OF THE MATERIAL FILE.
C
EXTERNALS: INBUF.
C
C
DATE: 13 DECEMBER 1976
C
C****
LCM Z,XTB
DIMENSION Z(I),XTB(J)
J = 5
K = XTB(J)
DO 1 I=1,K
IF(CTNR.EQ.XTB(J+I)) GO TO 3
1 CONTINUE
2 NUM = 0
RETURN
3 NUM = XTB(J+I+K)
IF(NUM.GT.NMAX) GO TO 6
DO 4 K=1,1
CALL INBUF(LU,Z,1,IEOF)
IF(IEOF.EQ.0) GO TO 2
4 CONTINUE
CALL INBUF(LU,Z,NUM,IEOF)
IF(IEOF.EQ.0) GO TO 2
I = I+1
DO 5 K=1,1
5 BACKSPACE LU
RETURN
6 NUM = NMAX-NUM
RETURN
END

SUBROUTINE RDCARD (LU,TBL,TBLID,NWDS,NWPC,FMT)
C PURPOSE ROUTINE READS BCD CARD IMAGE DATA FROM LOGICAL
C UNIT -LU-, DECIDES WHETHER HOLLERITH OR NUMERIC.
C AND IF NUMERIC, IT DECIDES WHETHER INTEGER OR
C FLOATING POINT.
C C ARGUMENTS
C LU (IN) - LOGICAL UNIT OF BCD INPUT FILE
C TBL (OUT) - SCM ARRAY TO HOLD DATA
C TBLLD (IN) - TABLE ID NUMBER
C NWDS (IN) - NUMBER OF WORDS TO BE READ
C NWPC (IN) - NUMBER OF WORDS PER CARD IMAGE
C FMT (IN) - BCD FORMAT (BAID=CCD, 20A4=IBM)
C C REMARKS IF TBLLD.LE.19999 DATA IS ASSUMED HOLLERITH, AND
C =NWDS= IS NUMBER OF HOLLERITH CHARACTERS TO BE READ.
C IF TBLLD.GE.20000 DATA IS ASSUMED NUMERIC. FIVE
C INTEGER FLAGS ARE READ WITH EACH CARD TO INDICATE
C TYPE OF EACH CORRESPONDING NUMBER. ALL DATA
C NUMBERS ARE READ WITH E15.0 FORMAT, BUT IF THE
C INTEGER FLAG SAYS IT SHOULD BE INTEGER TYPE, IT
C IS CONVERTED AND STORED BACK IN ITSELF.
C C PROGRAMMER C. FOREST, G. ROOD, T-4
C C DATE JAN 1977
C C------------------------------------------------------------------------
C DIMENSION TBL(1),TMP(5),IMP(S),IFLG(S)
C EQUIVALENCE (TMP,IMP)
C EQUIVALENCE(RKR,IRKR),(RKP,IRKP)
C IF(TBLLD.GT.1999) GO TO 20
C C----- READ HOLLERITH DATA
C NCDS=((NWDS*NWPC-1))/NWPC
C J=1,NWPC
C DO 10 L=1,NCDS
C J=J+NWPC
C K=J+NWPC-1
C IF(K.GT.NWDS) K=NWDS
C READ (LU,1000) (TBL(I),I=J,K)
C 10 FORMAT(20A4)
C 10 CONTINUE
C RETURN
C C----- READ NUMERIC TABLE
C 20 M=-5
C NCARD=(NWDS+5)/5
C DO 30 I=1,NCARD
C M=M+5
C READ (LU,100) (TMP(K),K=1,5), (IFLG(K),K=1,5)
C DO 30 J=1,5
C IF(IFLG(J),EQ,2) IMP(J)=TMP(J)
C TBL(J+M)=TMP(J)
C 30 CONTINUE
C RETURN
C 100 FORMAT(5E15.0,5I1)
C 110 FORMAT(5E15.0)
C END

***EOF*** 20 PAGES COST $3.00  PROJ BALANCE $10510.14  USER BALANCE $92.40
PARAMETER NREG=20,
    NDAT=5,
    ICDIM=32*NDAT,
    NCFW=NREG*NDAT
C TEST PROGRAM FOR SESAME LIBRARY

C VARIABLES:
C T  TEMPERATURE VECTOR
C D  DENSITY VECTOR
C P  PRESSURE VECTOR
C E  SPECIFIC INTERNAL ENERGY VECTOR
C DPDR DENSITY DERIVATIVE OF PRESSURE VECTOR
C DPDT TEMPERATURE DERIVATIVE OF PRESSURE VECTOR
C DEDR DENSITY DERIVATIVE OF SPECIFIC INTERNAL ENERGY
C DEDT TEMPERATURE DERIVATIVE OF SPECIFIC INTERNAL ENERGY
C Z  CHARGE STATE VECTOR
C SP PLANCK OPACITY VECTOR
C SR ROSSELAND OPACITY VECTOR
C UCONV UNIT CONVERSION FACTOR MATRIX
C IMAT MATERIAL INDEX VECTOR
C MATS SESAME MATERIAL ID NUMBER INPUT MATRIX

DIMENSION MATS(2,20)
  INCLUDE 'SESPAR.CMN'

DIMENSION IMAT(20), T(20), D(20), P(20), DPDR(20), DPDT(20),
1       E(20), DEDR(20), DEDT(20), Z(20), SR(20),
2       SP(20), XLOG(20), YLOG(20), UCONV(3,5)

C DEFINE THE VECTOR TO HOLD THE SESAME LIBRARY
REAL*8 A
COMMON/SESDAT/ A(15000)
REAL*4 B(1)
EQUIVALENCE (A(1),B(1))

C DEFINE THE COMMON BLOCKS FOR INPUT AND OUTPUT TO THE INTERPOLATION
C SUBROUTINE, NOTE THE DENSITY AND TEMPERATURE MUST BE INPUT AS
C DOUBLE PRECISION
REAL*8 X, V
COMMON/ESISIN/ IR, IDT, X, Y, IBR, IFL, IH
COMMON/ESISOUT/ ZI(3), ZZ(3)
COMMON/INTORD/ IFN
COMMON/RTBLK2/ LOCX, IX, NX, LOCY, IV

C SET UP THE MATERIAL ID'S FOR BERYLLIUM
MATS(1,1) = 2020
MATS(2,1) = 12020

C DEFINE THE IO UNIT NUMBERS
I6 = 16
LES = 3
LGP = 3
LOUT = 16

C SET THE LONG OUTPUT SWITCH TO ON
LONGIO = 1
C DEFINE THE LENGTH OF THE VECTOR TO HOLD THE SESAME DATA AND SET THE
C POINTER TO THE FIRST ELEMENT
LEN = 15000
LCNT = 1
C SET THE OPACITY OPTION TO GET OPACITIES AND SET THE INTERPOLATION
C OPTION TO USE THE RATIONAL FUNCTION APPROXIMATION
IRAD = 1
IFN = 0
C SET THE UNIT CONVERSION FACTORS TO SI UNITS
C G/CM3 --> KG/M3
C K --> K
C GJ/M3 --> PA
C GJ/MG --> J/KG

UCONV(1,1) = 1.E3
UCONV(2,1) = 1.E0
UCONV(3,1) = 1.E9
UCONV(3,2) = 1.E6
C G/CM3 --> KG/M3
C EV --> K
C -- --> --
UCONV(1,3) = 1.E3
UCONV(2,3) = 11604.E0
UCONV(3,3) = 1.E0
C G/CM3 --> KG/M3
C EV --> K
C CM2/G --> M2/KG

UCONV(1,4) = 1.E3
UCONV(2,4) = 11604.E0
UCONV(3,4) = 0.1E0
UCONV(3,5) = 0.1E0
C RETRIEVE THE SESAME DATA
CALL GETLIB(LEN,LCNT,NMATS,MATS,NWDS,SEES,UCONV,
    IRAD,LES,LOP,LOUT,LONGIO)
C INTERPOLATE FOR 20 (DENSITY,TEMPERATURE) POINTS
D(1) = 1.845E-2
D(2) = 1.845E-1
D(3) = 1.845
D(4) = 18.45
T(1) = 11604.E-2
T(2) = 11604.E-1
T(9) = 11604.
T(13) = 11604.E1
T(17) = 11604.E2
KMAX = 20
DO 100 K = 1,KMAX
   IMAT(K) = 1
100 CONTINUE

DO 150 K = 1,KMAX,4
   D(4+K) = D(1)
   D(5+K) = D(2)
   D(6+K) = D(3)
   D(7+K) = D(4)
150 CONTINUE

DO 200 K = 1,3
   T(K+K) = T(K)
   T(5+K) = T(5)
   T(9+K) = T(9)
   T(13+K) = T(13)
   T(17+K) = T(17)
200 CONTINUE

CALL EOSTAB(IMAT,T,D,P,DPDR,DPDT,E,DEDR,DEDT,
1                 S,Z,SR,SP,XLOG,YLOG,KMAX,IRAD)

WRITE(*,3000) (I, D(I), T(I), P(I), DPDR(I), DPDT(I),
1                 E(I), DEDR(I), DEDT(I),
1                 Z(I), SR(I), SP(I),
1                 I=1,KMAX)

3000 FORMAT('O'/O/30x,'INTERPOLATION OUTPUT'/O/
1                 'DENSITY TEMPERATURE PRESSURE DP/DD DP/DT ',
1                 'ENERGY DE/DD DE/DT ',
1                 'CHARGE ROSS OP PLANCK OP'/
1                 (1X,13,1P11E10.3))

STOP
END

SUBROUTINE CHKNEG(TBLS,LOC)
C---------------------------------------------------------------
C CHKNEG SEARCHES A 2-DIMENSIONAL SESAME TABLE FOR NEGATIVE
C DEPENDENT VARIABLES IN EITHER HALF-WORD. IF FOUND, THEY
C ARE SET TO ZERO.
C
C ARGUMENTS: TBLS (I/O) = ARRAY IN WHICH TABLES ARE LOADED
C            LOC (IN) = STARTING POINT IN ARRAY OF DESIRED TABLE.
C
C MODIFIED BY: G.A. MOSES FOR IMPLEMENTATION ON IBM HARDWARE
C FOR INR-KFK
C---------------------------------------------------------------
C
C REAL*8 TBLS(*)
C
C DDATA AND SDATA(1) AND SDATA(2) ALLOW ACCESS TO EACH HALF OF THE
C DOUBLE PACKED SESAME DATA ITEM
C REAL*8 DDATA
C REAL*8 SDATA(2)
C EQUIVALENCE (DDATA,SDATA(1))
C
C DETERMINE LOCATION AND NUMBER OF DEPENDENT VARIABLES
C NR=TBLS(LOC+2)
C NT=TBLS(LOC+3)
C NWDS=NR*NT
L=LOC+4+NR+NT

C NOW GO THRU THEM
DO 100 I=1,NWDS
   X1=0.
   X2=0.
C PUT THE LEFT HALF IN X1, THE RIGHT IN X2
DDATA = TBLS(L)
   X1 = SDATA(1)
   X2 = SDATA(2)
C CHECK THEM OUT, SET TO ZERO IF NEED BE
   IF (X1.LT.0.) X1=0.
   IF (X2.LT.0.) X2=0.
C REPACK THEM AND MOVE ON
   SDATA(1) = X1
   SDATA(2) = X2
TBLS(L) = DDATA
100   L=L+1
C RETURN
END

SUBROUTINE EOSTAB(IMAT,T,D,P,DPDR,DPDT,E,DED,DED,
   Z,SR,SP,XLOG,YLOG,KMAX,IRAD)

EOSTAB FORMS A LINK BETWEEN SESAME-II INTERPOLATION Routines
AND A CALLING PROGRAM SUCH AS SIMMER OR MEDUSA-KA

ARGUMENTS:

   IMAT  (I/O)  ABS = MATERIAL NUMBER VECTOR
             (= SESAME REGION NUMBER)
   T     (INPUT)  TEMPERATURE VECTOR
   D     (INPUT)  DENSITY VECTOR
   P     (OUTPUT) PRESSURE VECTOR
   DPDR  (OUTPUT) PRESSURE DENSITY DERIVATIVE VECTOR
   DPDT  (OUTPUT) PRESSURE TEMP DERIVATIVE VECTOR
   E     (OUTPUT) SPECIFIC INTERNAL ENERGY VECTOR
   DED   (OUTPUT) SPEC. INT. ENERGY DENs. DERIV. VECTOR
   DEDT  (OUTPUT) SPEC. INT. ENERGY TEMP. DERIV. VECTOR
   Z     (OUTPUT) CHARGE STATE VECTOR
   SR    (OUTPUT) ROSELLAND OPACITY VECTOR
   SP    (OUTPUT) PLANCK OPACITY VECTOR
   XLOG  (OUTPUT) TEMPORARY FOR LOGARITHM OF D
   YLOG  (OUTPUT) TEMPORARY FOR LOGARITHM OF T
   KMAX  (INPUT)  NUMBER OF VECTOR ELEMENTS TO PROCESS
   IRAD  (INPUT)  SWITCH FOR RADIATION CALCULATION
                   OPACITY SWITCH
                   1=CALCULATE OPACITY
                   0=DON'T

THE SESAME INTERPOLATION ROUTINES COMMUNICATE THROUGH THE SESIN AND
SESOUT COMMON BLOCKS

IR  MATERIAL NUMBER (1-10)
IDT DATA TYPE 1=P&E, 2=Z, 3=SR&SP
X  DENSITY VALUE FOR INTERPOLATION
Y  TEMPERATURE VALUE FOR INTERPOLATION
IBR 0=RETURN DATA FROM BOTH HALVES OF
DOUBLE PACKED DATA (I.E. P&E SR&SP)
1=RETURN DATA FROM LEFT HALF OF
DOUBLE PACKED DATA (I.E. Z)
IFL  ERROR FLAG
IH  GROUP NUMBER FOR MULTIGROUP OPACITIES
(CURRENTLY NOT USED)
Z1(1)  LEFT HALF DATA VALUE
Z1(2)  DENSITY DERIVATIVE OF LEFT HALF DATA
Z1(3)  TEMP. DERIVATIVE OF LEFT HALF DATA
Z2(1)  RIGHT HALF DATA VALUE
Z2(2)  DENSITY DERIVATIVE OF RIGHT HALF DATA
Z2(3)  TEMP. DERIVATIVE OF RIGHT HALF DATA

REAL*8 X, Y
COMMON/SESIN/IR,IDT,X,Y,IBR,IFL,IH
COMMON/SESOUT/Z1(3),Z2(3)

DECLARATION OF DUMMY ARGUMENTS
REAL*4 D(KMAX), T(KMAX), P(KMAX), DPDR(KMAX), DPDT(KMAX),
E(KMAX), DEDR(KMAX), DEDT(KMAX), Z(KMAX), SR(KMAX),
SP(KMAX), XLOG(KMAX), YLOG(KMAX)
INTEGER*4 IMAT(KMAX)

FORMATS
900 FORMAT( ' OVERFLOW/UNDERFLOW:', 4X, 2E10.5)
910 FORMAT( ' PASSED TO EOSTAB:', I3, 2E12.6)
920 FORMAT( ' SESIN:', 2I3, 2E12.6, 2I4)
925 FORMAT( ' SESOUT:', 2I3, 6E12.6)
930 FORMAT( ' EOSTAB RETURNS:', 5/(5014.5))

GET PRESSURE AND SPECIFIC ENERGY AND THEIR DERIVATIVES
IM = 0
IDT = 1
IBR = 0
DO 100 K = 1,KMAX
IR = IABS(IMAT(K))
X = D(K)
Y = T(K)
CALL T4DAT
P(K) = Z1(1)
DPDR(K) = Z1(2)
DPDT(K) = Z1(3)
E(K) = Z2(1)
DEDR(K) = Z2(2)
DEDT(K) = Z2(3)
100 CONTINUE

GET THE CHARGE STATE
THE SESAME CHARGE STATE DATA IS TABULATED LOGARITHMICALLY, SO THE
ACTUAL DENSITY AND TEMPERATURES MUST BE CONVERTED TO THEIR LOGS
BEFORE INTERPOLATION AND THE RESULT MUST BE RAISED TO THE POWER 10
TO GET THE ACTUAL CHARGE STATE VALUE
IBR = 1
IDT = 2
DO 200 K = 1,KMAX
IR = IABS(IMAT(K))
XLOG(K) = LOG10(D(K))
YLOG(K) = MAX(LOG10(T(K)), -3.)
X = XLOG(K)
Y = YLOG(K)
CALL T4DAT
Z(K) = 10. ** Z1(1)

200 CONTINUE

C GET THE ROSSELAND AND PLANCK OPACITIES IF REQUIRED

C THE ROSSELAND AND PLANCK OPACITIES ARE ALSO TABULATED LOGARITHMICALLY

IF ( IRAD .EQ. 0 ) GOTO 999

IBR = 0
IDT = 3
DO 300 K = 1,KMAX
   IR = IABS(IMAT(K))
   X = XLOG(K)
   Y = YLOG(K)
   CALL T4DAT
   SR(K) = 10. ** Z1(1)
   SP(K) = 10. ** Z2(1)
300 CONTINUE

C 999 RETURN
END

SUBROUTINE GETELC(IR, MID, IDT, TBLS, LCNT, LU, IFL, UCONV)

C*
C GETELC LOADS THE ELECTRON EOS TABLE
C*
C ARGUMENTS
C IR      (INPUT)  REGION NO.
C MID     (INPUT)  MATERIAL ID
C IDT     (INPUT)  DATA TYPE INDICATOR
C TBLS    (OUTPUT) ARRAY FOR TABLE STORAGE
C LCNT    (I/O)    POSITION IN ARRAY FOR STORING TABLES
C LU      (INPUT)  SESAME LIBRARY UNIT NO.
C IFL     (OUTPUT) ERROR FLAG
C = 2 FOR MATERIAL ALREADY LOADED
C = 1 FOR SUCCESSFUL LOADING
C = 0 FOR DATA NOT FOUND
C = - NO. OF EXTRA WORDS NEEDED FOR STORAGE
C UCONV   (INPUT)  UNIT CONVERSION MATRIX
C
C PRESSURES AND ENERGIES ARE DOUBLE PACKED
C
C WRITTEN BY  G.A. MOSES, UNIV. OF WISCONSIN, 3 JUNE 86
C
C COMMENT
C MODIFIED TO DOUBLE PACK DATA INTO DOUBLE PRECISION TBLS
C TO RUN ON IBM HARDWARE, MODELLED AFTER THE GETNUC
C ROUTINE. NO GETELC ROUTINE WAS INCLUDED WITH THE
C SESAME PACKAGE FROM LANL
C*
C*******************************************************************************
C
REAL*8 TBLS(1)
REAL*4 UCONV(3,5)
INCLUDE 'SESPLR,CMN'
COMMON /S2DIR/, LCXX, NRS, LCFW(NREG, NDAT)

C EQUIVALENCE IS USED FOR DOUBLE PACKING DATA
REAL*8 DDATA
REAL*4 SDATA(2)
EQUIVALENCE (DDATA,SDATA(1))

C UNIT CONVERSION FACTORS
RFAC = UCONV(1,1)
TFAC = UCONV(2,1)
PFAC = UCONV(3,1)
EFAC = UCONV(3,2)

CHECK TO SEE IF TABLE HAS BEEN LOADED
CALL MATCHK(MID,NRS,LCFW(1,1),TBLS(1),IFL)
IF(IFL.GT.0) GO TO 10
LCFW(IR,1) = IFL
IFL = 2
RETURN

10 NLEFT = LCMX - LCNT - 1

FETCH 304 TABLE
CALL TABSEQ(MID,304.,LU,TBLS(LCNT+2),NLEFT,IFL)
IF(IFL.LE.0) RETURN

CONVERT UNITS AND DOUBLE PACK
TBLS(LCNT) = DFLOAT(MID)
TBLS(LCNT+1) = DFLOAT(IDT)
NR = TBLS(LCNT+2)
NT = TBLS(LCNT+3)
NWDS = NR * NT
DO 20 I=1,NT
   LOCT = I + (LCNT + NR + 3)
20  TBLS(LOCT) = TFAC * TBLS(IOCT)

DO 40 J=1,NR
   TBLS(J+LCNT+3) = RFAC * TBLS(J+LCNT+3)
DO 30 I=1,NT
   LOCP = (I-1) * NR + J + (NT + NR + LCNT + 3)
30   LOCE = LOCP + NWDS
   PTEM = TBLS(LOCP) * PFAC
   ETEm = TBLS(LOCE) * EFAC
   SDATA(1) = PTEM
   SDATA(2) = ETEm
   TBLS(LOCP) = SDATA
40 CONTINUE

RESET INPUT PARAMETERS AND END
LCFW(IR,1) = LCNT
LCNT = LCNT + 2 * IFL - NWDS
IFL = 1
RETURN

END

SUBROUTINE GETEOS(IR,MID,IDT,TBLS,LCNT,LU,IFL,ZB,UCONV)

GETEOS LOADS THE TOTAL EOS (PRESSURE AND SPECIFIC INTERNAL ENERGY)
TABLES FOR A GIVEN MATERIAL AND DOUBLE PACKS THE DATA INTO THE
SESAME DATA VECTOR

ARGUMENTS:  IR (INPUT)  REGION NO.
            MID (INPUT)  SESAME MATERIAL ID.
            IDT (INPUT)  DATA TYPE INDICATOR  
            1=P&E, 2=Z, 3=SR&SP
            TBLS (IN/OUT) ARRAY FOR TABLE STORAGE
            LCNT (IN/OUT) POSITION IN ARRAY FOR STORING TABLES
            LU (IN/OUT)  SESAME LIBRARY IO UNIT NUMBER
            IFL (OUTPUT) ERROR FLAG
            = 2 FOR MATERIAL ALREADY LOADED
C = 1 FOR SUCCESSFUL LOADING
C = 0 FOR DATA NOT FOUND
C = - NO. OF EXTRA WORDS NEEDED FOR STORAGE
C
C ZB (OUTPUT) ATOMIC CHARGE, CHARGE**2, AND MASS
C ZB(1) = Z
C ZB(2) = Z**2
C ZB(3) = A
C
C UNITS PRESSURE (JERKS/CC)
C ENERGY DENSITY (JERKS/GM)
C TEMPERATURE (KEV)
C DENSITY (GM/CC)
C
C MODIFIED BY: G.A. MOSES FOR IMPLEMENTATION ON IBM HARDWARE
C FOR INR-KFK MAY 1986
C
C*****************************************************************************
C
C INCLUDE 'SESPAR.CMN'
COMMON/S2DIR/ LCMX, NRS, LCFW(NREG, NDAT)
REAL*8 TBLS(*), ZB(3), RHO0
REAL*4 UCONV(3,5)
C
C DDATA AND SDATA(1) AND SDATA(2) ARE USED TO ACCESS SESAME DATA
C DOUBLE PACKED INTO AN IBM DOUBLE WORD
REAL*8 DDATA
REAL*4 SDATA(2)
EQUVALENCE (DDATA, SDATA(1))
C
C SET THE UNIT CONVERSION FACTORS
RFAC = UCONV(1,1)
TFAC = UCONV(2,1)
PFAC = UCONV(3,1)
EFAC = UCONV(3,2)
C
C CHECK TO SEE IF TABLE HAS BEEN LOADED
CALL MATCHK(MID, NRS, LCFW(1, IDT), TBLs(1), IFLG)
IF(IFLG .EQ. 0) GO TO 10
LCFW(IR, IDT) = IFLG
IFL = 2
RETURN
10 NLEFT = LCMX - LCNT - 1
C
C FETCH THE 201 TABLE
CALL TABSEQ(MID, 201, LU, TBLs(LCNT+2), NLEFT, IFL)
IF(IFL .LE. 0) RETURN
ZB(1) = TBLs(LCNT+2)
ZB(2) = ZB(1)*ZB(1)
ZB(3) = TBLs(LCNT+3)
RHO0 = TBLs(LCNT+4) * RFAC
C
C FETCH THE 301 TABLE
CALL TABSEQ(MID, 301, LU, TBLs(LCNT+2), NLEFT, IFL)
IF(IFL .LE. 0) RETURN
C
C CONVERT TO DESIRED UNITS AND DOUBLE PACK
NR = TBLs(LCNT+2)
NT = TBLs(LCNT+3)
DO 20 I=1, NT
LOCT = I + (LCNT + NR + 3)
20 CONTINUE
CONVERT THE TEMPERATURE UNITS
20 TBLS(LOCT) = TFAC * TBLS(LOCT)
NWDS = NR * NT
DO 30 J = 1, NR

CONVERT DENSITY UNITS
TBLS(J+LCNT+3) = TBLS(J+LCNT+3) * RFAC
DO 40 I = 1, NT
LOCP = (I-1) * NR + J + (NT + NR + LCNT + 3)
LOCE = LOCP + NWDS

CONVERT THE PRESSURE FROM GPA TO JERKS/CC AND THE SPECIFIC
INTERNAL ENERGY FROM GJ/GM TO JERKS/GM
PTEM = TBLS(LOCP) * PFAC
ETEM = TBLS(LOCE) * EFAC

DOUBLE PACK THE DATA
SDATA(1) = PTEM
SDATA(2) = ETEM
TBLS(LOCP) = DDATA
40 CONTINUE
30 CONTINUE

RESET INPUT PARAMETERS AND END
TBLS(LCNT) = DFLOAT(MID)
TBLS(LCNT+1) = RH00
LCPW(IR,IDT) = LCNT
LCNT = LCNT + 2 + IFL - NWDS
IFL = 1
RETURN
END
SUBROUTINE GETLIB (LENA,LCNT,NMATS,MATS,NWDOSES,UCONV,
& IRAD,LES,LOP,LOUT,LONGIO)

GETLIB LOADS EQUATION OF STATE, CHARGE STATE, AND OPACITY TABLES
FROM THE SESAME II LIBRARY

ARGUMENTS:
LENA = (IN) LENGTH OF THE VECTOR TO HOLD THE
SESAME TABLES IN DOUBLE WORDS
LCNT = (IN) THE POSITION IN TBLS WHERE THE TABLES
ARE TO BE LOADED.
(OUT) THE LOCATION OF THE NEXT EMPTY WORD
IN TND.
THE SESAME 'GET...' ROUTINES USE THIS ARGUMENT
AND INCREMENT IT.
NMATS = (OUT) NUMBER OF MATERIALS LOADED
(*) NO. OF SESAME REGIONS
MATS = (IN) A SEQUENCE OF UP TO 20 2-PART
VECTORS WHOSE COMPONENTS DENOTE:
(1) SESAME MATERIAL NUMBER FOR EOS.
(2) " " " Z AND OPACITY.
NWDOSES = (OUT) NO. OF DOUBLE WORDS TAKEN UP
BY SESAME TABLES
IRAD = (IN) RADIATION & OPACITY SWITCH
0/1 = DON'T LOAD OPACITY TABLES/DO LOAD
I/O UNIT
LES = SESAME EQUATION-OF-STATE LIBRARY (P & E)
NUMBERS
LOP = SESAME OPACITY LIBRARY (Z, KR, & KP)
LOUT = OUTPUT FROM GETLIB
LONGIO = (IN) = 0 SHORT SUMMARY OF DATA TABLES
= 1 COMPLETE LISTING OF DATA

REMARKS
THE DATA CAN SUBSEQUENTLY BE REFERENCED BY SPECIFYING
TO A SESAME INTERPOLATION ROUTINE (VIA COMMON/SZDIR):
(1) THE MATERIAL REGION NUMBER, 1-20
(SESAME REGION NO., OR IR)
(2) THE DATA-TYPE INDEX, 1-3, WHERE
IDT=1 ==> P & E
2 ==> Z
3 ==> KR & KP

------------------------------------------------------------------------

INCLUDE 'SESPAR.CMN'
COMMON/SZDIR/ LCXM,NRS,LCFW(NREG,NDAT)

REAL*4 UCONV(3,5)
REAL*8 TBL(1)
COMMON/SESDAT/ TBL(1)

PARAMETER (LNXI = 1000, LNV1 = 1000)
COMMON/KFNDAT/ LNX(NDAT,NREG), LNX(NDAT,NREG),
1 LBX(NDAT,NREG), LBY(NDAT,NREG),
2 LPX(NDAT,NREG), LPY(NDAT,NREG),
3 NX(LNXI,NDAT,NREG), NY(LNV1,NDAT,NREG)

REAL*8 ZB(3)
DIMENSION MATS(2,NREG)

FORMAT 900 FORMAT(' ENTERING SUBROUTINE GETLIB. ')
910 FORMAT(' SUB GETLIB: NO EOS TABLE LOADED FOR MAT',I2, '#',I6,
& ', 'FLAG=',I7)
912 FORMAT(' SUB GETLIB: NO CHARGE-STATE TABLE LOADED FOR MAT',I2, '#'
& ', 'FLAG=',I7)
914 FORMAT(' SUB GETLIB: NO OPAQITY TABLE LOADED FOR MAT',I2, '#',I6,
& ', 'FLAG=',I7)
931 FORMAT(' SUB GETLIB: ERROR IN KFK FAST INTERPOLATION '
1 'INITIALIZATION. PROGRAM STOPS.')
932 FORMAT(1X,I5,1X,I10)
933 FORMAT('0'/ 'MATERIAL SESAME ID SESAME ID & E TABLE ',
1 'NUMBER OF NUMBER OF Z TABLE NUMBER OF ',
1 'NUMBER OF OP TABLE NUMBER OF NUMBER OF ',
2 'INDEX P & E Z & OPAC POINTER DENSITIES ',
2 'TEMPS POINTER DENSITIES TEMPS ')
934 FORMAT(1X,I3,1X,I12E10.3)
935 FORMAT('0'/
1 'MAT ---------PRESSURE AND ENERGY-------- ',
1 '---------CHARGE STATE------------- ',
1 '-----ROSELAND AND PLANCK OPAQITIES--- ',
2 'ID MIN DENS MAX DENS MIN TEMP MAX TEMP ',
2 'MIN DENS MAX DENS MIN TEMP MAX TEMP ')

WRITE SALUTARY MESSAGE
WRITE(LOUT,900)

COMMON /SZDIR/ MUST KNOW ITS OWN PARAMETER VARIABLES
LCMX = LENA
NRS = NREG

C
C INITIALIZE A FEW VARIABLES
LC1 = LCNT
NMATS = 0
DO 5 I = 1,10
DO 5 J = 1,3
LCFW(I,J) = 0
5 CONTINUE
C
C LOOP THRU THE MATERIAL VECTORS
C
100 DO 500 I = 1,NREG
   IF (MATS(I,1) .EQ. 0 .AND. MATS(2,I) .EQ. 0) GOTO 1050
C
C SESAME EQUATION OF STATE SUB-LIBRARY - P & E
MID=MATS(I,1)
   IF( MID .NE. 0 ) THEN
C
C LOAD P & E TABLE FOR IDT=1
   IDT = 1
   CALL GETEOS(I,MID,IDT,TBLS,LCNT,LES,IFL,ZB,UCONV)
   IF (IFL.LE.0) WRITE(OUTPUT,910) I,MID,IFL
   ENDFI
C
C SESAME ASTROPHYSICAL SUB-LIBRARY - Z & OPACITIES
MID=MATS(2,I)
   IF( MID .NE. 0 ) THEN
C
C LOAD CHARGE-STATE TABLE FOR IDT=2
   IDT = 2
   CALL GETMFE(I,MID,IDT,TBLS,LCNT,LOP,IFL,UCONV)
   IF (IFL.LE.0) WRITE(OUTPUT,912) I,MID,IFL
C
C IF (IRAD .NE. 0) THEN
C
C LOAD ONE TEMPERATURE OPACITY TABLE FOR IDT=3
   IDT = 3
   CALL GETRPO(I,MID,IDT,TBLS,LCNT,LOP,IFL,UCONV)
   IF (IFL .LE. 0) WRITE(OUTPUT,914) I,MID,IFL
   ENDFI
   ENDFI
C
C UPDATE THE MATERIAL COUNT
NMATS = NMATS + 1
C
C END OF LOOP THROUGH MATERIALS
500 CONTINUE
C
C NOW PROCESS THE DATA THAT IS READ INTO THE TABLES
1050 CONTINUE
C
C GET TABLE SIZE AND CHECK ON P OR E < 0.
NWSSES = LCNT - LC1
   DO 1500 I = 1,NMATS
   LOC = LCFW(I,1)
1500 CALL CHKNEG(TBLS,LOC)
C
C SET UP INITIALIZATION INFORMATION FOR KFK FAST INTERPOLATION
C
   DO 2800 I = 1,NMATS
DO 2800 J = 1,3
   LOC = LCFW(I,J)
   XX = TBLs(LOC+2)
   KY = TBLs(LOC+3)
   CALL INLIX(KX,KY,TBLs(LOC+4),TBLs(LOC+4+KX),KER,
           LNX(I,J),LNY(I,J),LXy(I,J),LCy(I,J),
           LPx(I,J),LPy(I,J),NX(I,J,Ny(I,J,Nx(I,J,LNXy,LNYy))
   IF( KER .NE. 0 ) THEN
      WRITE(LOUT,931)
      STOP
   ENDIF
2800 CONTINUE
SUMMARIZE THE DATA TABLES
   WRITE(LOUT,933)
   DO 2900 N = 1,NMATS
C GET POINTERS TO (1) P&E, (2) Z, AND (3) OPACITIES
   LOC1 = LCFW(N,1)
   LOC2 = LCFW(N,2)
   LOC3 = LCFW(N,3)
C GET NUMBER OF DENSITY AND TEMPERATURE INDICES FOR EACH DATA TYPE
   IF( LOC1 .NE. 0 ) THEN
      NR1 = TBLs(LOC1+2)
      NT1 = TBLs(LOC1+3)
   ENDIF
   IF( LOC2 .NE. 0 ) THEN
      NR2 = TBLs(LOC2+2)
      NT2 = TBLs(LOC2+3)
   ENDIF
   IF( IRAD .NE. 0 .AND. LOC3 .NE. 0 ) THEN
      NR3 = TBLs(LOC3+2)
      NT3 = TBLs(LOC3+3)
   ENDIF
   WRITE(LOUT,932) N,MATS(1,N),MATS(2,N),
           LOC1,NR1,NT1,
           LOC2,NR2,NT2,
           LOC3,NR3,NT3
2900 CONTINUE
   WRITE(LOUT,935)
   DO 2950 N = 1,NMATS
C GET POINTERS TO P&E(1), Z(2), AND OPACITIES(3)
   LOC1 = LCFW(N,1)
   LOC2 = LCFW(N,2)
   LOC3 = LCFW(N,3)
C GET NUMBER OF DENSITY AND TEMPERATURE INDICES FOR EACH DATA TYPE
   IF( LOC1 .NE. 0 ) THEN
      NR1 = TBLs(LOC1+2)
      NT1 = TBLs(LOC1+3)
   ENDIF
   IF( LOC2 .NE. 0 ) THEN
      NR2 = TBLs(LOC2+2)
      NT2 = TBLs(LOC2+3)
   ENDIF
   IF( IRAD .NE. 0 .AND. LOC3 .NE. 0 ) THEN
      NR3 = TBLs(LOC3+2)
      NT3 = TBLs(LOC3+3)
   ENDIF
C GET MIN AND MAX DENSITIES AND TEMPERATURES FOR EACH DATA TYPE
   IF( LOC1 .NE. 0 ) THEN
      D1LOW = TBLs(LOC1+4)
D1HI = TBLS(LOC1+4+NR1-1)
T1LOW = TBLS(LOC1+4+NR1)
T1HI = TBLS(LOC1+4+NR1+NT1-1)
ELSE
   D1LOW = -1.
   D1HI = -1.
   T1LOW = -1.
   T1HI = -1.
ENDIF
IF( LOC2 .NE. 0 ) THEN
   D2LOW = 10. ** TBLS(LOC2+4)
   D2HI = 10. ** TBLS(LOC2+4+NR2-1)
   T2LOW = 10. ** TBLS(LOC2+4+NR2)
   T2HI = 10. ** TBLS(LOC2+4+NR2+NT2-1)
ELSE
   D2LOW = -1.
   D2HI = -1.
   T2LOW = -1.
   T2HI = -1.
ENDIF
IF( IRAD .NE. 0 .AND. LOC3 .NE. 0 ) THEN
   D3LOW = 10. ** TBLS(LOC3+4)
   D3HI = 10. ** TBLS(LOC3+4+NR3-1)
   T3LOW = 10. ** TBLS(LOC3+4+NR3)
   T3HI = 10. ** TBLS(LOC3+4+NR3+NT3-1)
ELSE
   D3LOW = -1.
   D3HI = -1.
   T3LOW = -1.
   T3HI = -1.
ENDIF
WRITE(LOUT,934) N, D1LOW, D1HI, T1LOW, T1HI,
   D2LOW, D2HI, T2LOW, T2HI,
   D3LOW, D3HI, T3LOW, T3HI
2950 CONTINUE
C IF( LONG10 .EQ. 1 ) THEN
C WRITE THE FULL BLOWN DATA TABLES
   DO 2975 N = 1,NMATS
   C FIRST THE P&E TABLE
   LOC = LCFW(N,1)
   IF( LOC .NE. 0 ) THEN
      LR = 1
      IDT = 1
      CALL OUTTAB(LOC,IDT,LR,LOUT)
      LR = 2
      IDT = 1
      CALL OUTTAB(LOC,IDT,LR,LOUT)
   ENDIF
C NEXT THE Z TABLE
   LOC = LCFW(N,2)
   IF( LOC .NE. 0 ) THEN
      LR = 1
      IDT = 2
      CALL OUTTAB(LOC,IDT,LR,LOUT)
   ENDIF
C FINALLY THE OPACITY TABLE
LOC = LCFW(N,3)
IF( LOC .NE. 0 ) THEN
LR = 1
IDT = 3
CALL OUTTAB(LOC,IDT,LR,LOUT)
LR = 2
IDT = 3
CALL OUTTAB(LOC,IDT,LR,LOUT)
ENDIF
2975 CONTINUE
ENDF
C RETURN
END
SUBROUTINE GETNFE(IR,MID,IDT,TBLS,LCNT,LU,IFL,UCONV)
C---------------------------------------------------------------------
C GETNFE LOADS THE CHARGE STATE DATA INTO THE SESAME DATA VECTOR
C FOR A SPECIFIED MATERIAL
C
C ARGUMENTS:  IR   (INPUT)  REGION. NO.
C             MID  (INPUT)  SESAME MATERIAL ID
C             IDT  (INPUT)  DATA TYPE
C             TBLS (IN/OUT) TABLE STORAGE ARRAY
C             LCNT (IN/OUT) POSITION IN ARRAY FOR
C             LOADIND TABLES
C             LU   (INPUT)  SESAME LIBRARY ID UNIT NO.
C             IFL  (OUTPUT) 2=MATERIAL LOADED PREVIOUSLY
C                                1=MATERIAL LOADED SUCCESSFULLY
C                                0=MATERIAL NOT FOUND
C                                =- THE NUMBER OF EXTRA WORDS NEEDED
C                                FOR LOADING
C
C MODIFIED BY:  G.A. MOSES FOR IMPLEMENTATION ON IBM HARDWARE
C FOR INR-KFK MAY 1986
C---------------------------------------------------------------------
INCLUDE 'SESPAR,CMN'
COMMON/S2DIR//LMX,NRS,LCFW(NREG,NDAT)
REAL*8 TBLS(*)
REAL*4 UCONV(3,5)
C DDATA AND SDATA(1) AND SDATA(2) ARE USED TO ACCESS SESAME DATA THAT
C IS DOUBLE PACKED INTO AN IBM DOUBLE WORD
REAL*8 DDATA
REAL*4 SDATA(2)
EQUIVALENCE (DDATA,SDATA(1))
C CONVERSION FACTORS FOR THE LOGARITHMIC VALUES OF TEMPERATURE AND DENSITY
RFAC = LOG10(UCONV(1,3))
TFAC = LOG10(UCONV(2,3))
ZFAC = LOG10(UCONV(3,3))
C CHECK TO SEE IF THE DATA IS ALREADY LOADED
CALL MATCHK(MID,NRS,LCFW(1,IDT),TBLS(1),IFL)
IF(IFL .EQ. 0) GO TO 10
LCFW(IR,IDT) = IFL
IFL = 2
RETURN
10 NL = LCMX - LCNT - 1
C FETCH THE 504 TABLE
CALL TASEQ(MID,504,LU,TBL(LCNT+2),NL,IFL)
IF(IFL.LE.0) RETURN
NR = TBL(LCNT+2)
NT = TBL(LCNT+3)
IPT = LCNT + 3 + NR + NT
DO 20 I = 1,NT
C
C CONVERT THE TEMPERATURE UNITS
TBL(LCNT+3+NR+I) = TBL(LCNT+3+NR+I) + TFAC
DO 20 J = 1,NR
IPT = IPT+1
IF(I.GT.1) GO TO 15
C
C CONVERT THE DENSITY UNITS
TBL(LCNT+3+J) = TBL(LCNT+3+J) + RFAC
C
C PACK THE CHARGE STATE INTO THE LEFT HALF OF THE DOUBLE WORD
AND SET THE RIGHT HALF TO ZERO, WE DON'T CURRENTLY USE IT
15 ZF = TBL(IPT) + ZFAC
SDATA(1) = ZF
SDATA(2) = 0.0
TBL(IPT) = DDATA
C
C SAVE THE MATERIAL ID AND THE DATA TYPE AND A POINTER TO THE DATA
IN THE TABLE
TBL(LCNT) = DFLOAT(MID)
TBL(LCNT+1) = DFLOAT(IDT)
LCFW(IR,IDT) = LCNT
LCNT = LCNT + 2 + IFL
IFL = 1
C
RETURN
END
SUBROUTINE GETNUC(IR,MID,IDT,TBL,LCNT,LU,IFL,UCONV)

***********************************************************************
C GESTNUC LOADS THE NUCLEAR EOS TABLE
C
C ARGUMENTS
C IR (INPUT) REGION NO.
C MID (INPUT) MATERIAL ID
C IDT (INPUT) DATA TYPE INDICATOR
C TBLS (OUTPUT) ARRAY FOR TABLE STORAGE
C LCNT (I/O) POSITION IN ARRAY FOR STORING TABLES
C LU  (OUTPUT) SESAME LIBRARY UNIT NO.
C IFL (OUTPUT) ERROR FLAG
C = 2 FOR MATERIAL ALREADY LOADED
C = 1 FOR SUCCESSFUL LOADING
C = 0 FOR DATA NOT FOUND
C = - NO. OF EXTRA WORDS NEEDED FOR STORAGE
C UCONV (INPUT) UNIT CONVERSION MATRIX
C PRESSURES AND ENERGIES ARE DOUBLE PACKED
C
C MODIFIED G.A. MOSES, UNIV. OF WISCONSIN, 3 JUNE 86
C
C COMMENT MODIFIED TO DOUBLE PACK DATA INTO DOUBLE PRECISION TBLS
C TO RUN ON IBM HARDWARE
C
C***************************************************************
C
C            REAL*8 TBL(1)
C            REAL*4 UCONV(3,5)
C            INCLUDE 'SUSPAR.CMN'
C            COMMON / S2DIR / LCMAX,NRS,LCFW(NREG,NDAT)
C
C            EQUIVALENCE IS USED FOR DOUBLE PACKING DATA
C            REAL*4 DDATA
C            REAL*8 SDATA(2)
C            EQUIVALENCE (DDATA,SDATA(1))
C
C            UNIT CONVERSION FACTORS
C            RFAC = UCONV(1,1)
C            TFAC = UCONV(2,1)
C            PFAC = UCONV(3,1)
C            EFAC = UCONV(3,2)
C
C            CHECK TO SEE IF TABLE HAS BEEN LOADED
C            CALL MATCH(MID,NRS,LCFW(1,IDT),TBL(1),IFLAG)
C            IF(IFLAG .EQ. 0) GO TO 10
C            LCFW(I,N) = IFLAG
C            IFL = 2
C            RETURN
C
C            10 NLEFT = LCMAX + LCNT - 1
C
C            FETCH 303 TABLE
C            CALL TABSEQ(MID,303,LU,TBL(LCNT+2),NLEFT,IFL)
C            IF(IFLAG .LE. 0) RETURN
C
C            CONVERT UNITS AND DOUBLE PACK
C            TBL(LCNT) = DFLOAT(MID)
C            TBL(LCNT+1) = DFLOAT(IDT)
C            NR = TBL(LCNT+2)
C            NT = TBL(LCNT+3)
C            NWDS = NR * NT
C            DO 20 I = 1,NT
C            LOC1 = I + (LCNT + NR + 3)
C            20 TBL(LOC1) = RFAC * TBL(LOC1)
C            DO 40 J = 1, NR
C            TBL(J + LCNT + 3) = RFAC * TBL(J + LCNT + 3)
C            DO 30 I = 1, NT
C            LOC2 = (I-1) * NR + J + (NT + NR + LCNT + 3)
C            LOCE = LOC2 + NWDS
C            PTERM = TBL(LOC2) * PFAC
C            ETTERM = TBL(LOCE) * EFAC
C            SDATA(1) = PTERM
C            SDATA(2) = ETTERM
C            TBL(LOC2) = DDATA
C            30 CONTINUE
C            40 CONTINUE
C
C            RESET INPUT PARAMETERS AND END
C            LCFW(I,N) = LCNT
C            LCNT = LCNT + 2 * IFL - NWDS
C            IFL = 1
C            RETURN
C            END
C
C            SUBROUTINE GETRPO(IR,MID,IDT,TBL,LCNT,LU,IFL,UCONV)
C GETRPO LOADS THE ROSELAND AND PLANCK OPACITY DATA FOR A SPECIFIED
C MATERIAL INTO THE SESAME DATA VECTOR
C
C ARGUMENTS: IR (INPUT) REGION NO.
C MID (INPUT) SESAME MATERIAL ID
C IDT (INPUT) DATA TYPE INDICATOR
C 1=P&E, 2=Z, 3=SR&SP
C TBL5 (IN/OUT) ARRAY FOR TABLE STORAGE
C LCNT (IN/OUT) POSITION IN ARRAY FOR
C STORING TABLES
C LU (INPUT) SESAME LIBRARY ID UNIT NO.
C IFL (OUTPUT) ERROR FLAG
C =2 TABLE ALREADY LOADED
C =1 FOR SUCCESSFUL LOADING
C =0 FOR DATA NOT FOUND
C =-NO. OF EXTRA WORDS NEEDED
C TO STORE DATA
C
C MODIFIED BY: G.A. MOSES FOR IMPLEMENTATION ON IBM HARDWARE FOR
C INR-KFKA MAY 1986
C
C---------------------------------------------------------------------------
C INCLUDE 'SESPAR.CMN'
C COMMON/S2DIR/ LCMX,NRS,LCFW(NREG,NDAT)
C
C DDATA AND SDATA(1) AND SDATA(2) ARE USED TO ACCESS SESAME DATA THAT IS
C DOUBLE PACKED INTO AN IBM DOUBLE WORD
C REAL*8 TBL5(*),
C REAL*4 UCONV(3,5)
C REAL*8 DDATA
C REAL*4 SDATA(2)
C EQUIVALENCE (DDATA,SDATA(1))
C
C UNIT CONVERSION FACTORS, REMEMBER THE DATA IS LOGARITHMS
C RPAC = LOG10(UCONV(1,4))
C TFAC = LOG10(UCONV(2,4))
C ROFAC = LOG10(UCONV(3,4))
C POFAC = LOG10(UCONV(3,5))
C
C CHECK TO SEE IF TABLE HAS BEEN LOADED ALREADY
C CALL MATCHK(MID,NRS,LCFW(1,IDT),TBL5(1),IFL)
C IF(IFL.EQ.0) GO TO 10
C LCFW(IR,IDT) = IFL
C IFL = 2
C RETURN
C 10 NLEFT = LCMX - LCNT + 1
C
C FETCH THE 502 TABLE - ROSELAND OPACITY
C CALL TABSEQ(MID,502,LU,TBL5(LCNT+2),NLEFT,IFL)
C IF(IFL.LT.0) RETURN
C
C FETCH THE 505 TABLE - PLANCK OPACITY
C NLEFT = NLEFT - IFL
C CALL TABSEQ(MID,505,LU,TBL5(LCNT+2+IFL),NLEFT,IFLP)
C IF(IFLP.GT.0) GOTO 30
C IFL = IFLP
C RETURN
C
C CONVERT TO DESIRED UNITS AND DOUBLE PACK
C 30 NR = TBL5(LCNT+2)
C NT = TBL5(LCNT+3)
IPT = LCNT + 3 + NR + NT
DO 60 K = 1, NT
TBLS(LCNT+3+NR*K) = TBLS(LCNT+3+NR*K) + TFAC
DO 60 J = 1, NR
IF(K .GT. 1) GO TO 50
TBLS(LCNT+3+J) = TBLS(LCNT+3+J) + RFAC
50 IPT = IPT+1
C
C DOUBLE PACK THE ROSSELAND OPACITY INTO THE LEFT HALF AND
C PLANCK OPACITY INTO THE RIGHT HALF
SDATA(1) = TBLS(IPT)
SDATA(2) = TBLS(IPT+IFL)
SDATA(1) = SDATA(1) + ROFAC
SDATA(2) = SDATA(2) + POFAC
TBLS(IPT) = DDATA
60 CONTINUE
C
C SAVE THE MATERIAL ID AND THE DATA TYPE AND THE POINTER TO THIS
C DATA TABLE
TBLS(LCNT) = DFLOAT(MID)
TBLS(LCNT+1) = DFLOAT(IDT)
LCFW(IR,IDT) = LCNT
LCNT = LCNT + IFL + 2
IFL = 1
C
RETURN
END
SUBROUTINE GETRL(IR,MID,IDT,ITOT,TBLS,LCNT,LU,IFL,UCONV)
C
GETRL LOADS THE ROSSELAND/TOTAL OPACITY TABLE
C
ARGUMENTS
C
IR (INPUT) REGION NO.
C MID (INPUT) MATERIAL ID.
C IDT (INPUT) DATA TYPE INDICATOR
C ITOT (INPUT) 0=LOAD ROSSELAND TABLE ONLY
C 1=LOAD BOTH ROSSELAND AND TOTAL
C TABLES
C TBLS (OUTPUT) ARRAY FOR TABLE STORAGE
C LCNT (IN/OUT) POSITION IN ARRAY
C FOR STORING TABLES
C LU (INPUT) SESAME II LIBRARY UNIT NO.
C IFL (OUTPUT) ERROR FLAG
C =2 FOR MATERIAL PREVIOUSLY LOADED
C =1 FOR MATERIAL LOADED
C =0 FOR DATA NOT FOUND
C =-THE NO. OF EXTRA WORDS NEEDED
C UCONV (INPUT) UNIT CONVERSION FACTORS
C
REMARKS
THE ROSSELAND TABLE IS Fetched. UNITS CONVERTED, AND
DOUBLED PACKED INTO BOTH THE FIRST AND SECOND HALFWORDS.
IF ITOT=1 THE ELECTRON CONDUCTION OPACITY TABLE
IS Fetched. UNITS CONVERTED. THE TOTAL OPACITY
IS THEN COMPUTED AND ITS VALUE REPLACES THAT
VALUE STORED IN THE SECOND HALFWORD.
C
MODIFIED G.A. MOSE, UNIV. OF WISCONSIN, 3 JUNE 86
C
COMMENT DATA IS DOUBLE PACKED IN DOUBLE PRECISION TBLS
C TO RUN ON IBM HARDWARE
C
REAL*8 TBLS(1)
REAL*4 UCONV(3,5)
INCLUDE 'SESPAR.CMN'
COMMON/S2DIR/LCMX,NRS,LCFW(NREG,NDAT)
C
C EQUIVALENCE IS USED FOR PACKING AND UNPACKING DATA
REAL*8 DDATA
REAL*4 SDATA(2)
EQUIVALENCE (DDATA,SDATA(1))
C
C UNIT CONVERSION
RFAC = LOG10(UCONV(1,4))
TFAC = LOG10(UCONV(2,4))
ROFAC = LOG10(UCONV(3,4))
TOFAC = LOG10(UCONV(3,5))
C
C CHECK TO SEE IF MATERIAL IS ALREADY LOADED
CALL MATCHK(MID,NRS,LCFW(1,1DT),TBLS(1),IFL)
IF(IFL.EQ.0)GO TO 10
LCFW(IR,IDT) = IFL
IFL = 2
RETURN
C
C GET THE ROSSELAND TABLE
10 NL = LCMX - LCNT - 1
CALL TABSEQ(MID,502.,LU,TBLS(LCNT+2),NL,IFL)
IF(IFL.LE.0)RETURN
C
C CONVERT UNITS
NR = TBLS(LCNT+2)
NT = TBLS(LCNT+3)
IPT = LCNT+3+NR+NT
DO 100 I=1,NT
TBLS(LCNT+3+NR+I) = TBLS(LCNT+3+NR+I) + TFAC
DO 100 J=1,NR
IF(J.GT.1)GO TO 30
TBLS(LCNT+3+J) = TBLS(LCNT+3+J) + RFAC
30 IPT = IPT + 1
ROP = TBLS(IPT) + ROFAC
SDATA(1) = ROP
SDATA(2) = ROP
TBLS(IPT) = DDATA
100 CONTINUE
C
C IF TOTAL OPACITY IS NOT NEEDED FINISH UP
IF(ITOT.EQ.0)GO TO 500
C
C FETCH THE ELECTRON COND. OPACITY TABLE
NL = NL = IFL
CALL TABSEQ(MID,503.,LU,TBLS(LCNT+2+IFL),NL,IFLE)
IF(IFLE.GT.0)GO TO 120
IFL = IFLE
RETURN
C
C ADD TABLES TO GET TOTAL OPACITY
120 NTE = TBLS(LCNT+3+IFL)
LE = LCNT+3+IFL+NR+NTE
IPT = LCNT+3+NR+NT
DO 200 I=1,NT
IF(I.GT.NTE)GO TO 500
DO 200 J=1,NR
IPT = IPT + 1
LE = LE + 1
TOFAC = TBLS(LE) + TOFAC
ROP = TBLS(IPT)
TOP = 1.0 / 10.0**TOP + 1.0 / 10.0**ROP
TOP = LOG10(1.0/TOP)
SDATA(1) = ROP
SDATA(2) = TOP
TBLS(IPT) = DDATA

200 CONTINUE
500 LCFW(IR,IDT) = LCNT
TBLS(LCNT) = OFLOAT(MID)
TBLS(LCNT+1) = OFLOAT(IDT)
LCNT = LCNT+2+IFL
IFL = 1
RETURN
END

SUBROUTINE GETTCE(IR,MID,IDT,TBLS,LCNT,LU,IFL)

GETTCE LOADS TABLES OF ELECTRONIC THERMAL CONDUCTIVITIES
FROM SESAME II

ARGUMENTS
IR  (INPUT)  REGION NO.
MID (INPUT)  MATERIAL ID
IDT (INPUT)  DATA TYPE INDICATOR
TBLS (INPUT)  STORAGE ARRAY FOR TABLE
LCNT (INPUT)  POSITION IN ARRAY FOR LOADING TABLE
LU  (INPUT)  SESAME II LIBRARY UNIT NO.
IFL (OUTPUT)  =1 FOR SUCCESSFUL LOADING
              0 FOR DATA NOT FOUND
              - THE NO. OF EXTRA WORDS NEEDED
              FOR LOADING

REMARKS
THIS VERSION OF GETTCE IS SPECIALLY DESIGNED FOR USE
IN LASNEX. IF NECESSARY THIS ROUTINE MAY BE MODIFIED
FOR ADAPTATION TO OTHER CODES.

EXTERNALS
MATCHK,TABFCH,DPACK

PROGRAMMER  J.ABDALLAH, JR.

DATE  24 APRIL 1979

MODIFIED  G.A. MOSES. UNIV. OF WISCONSIN, 3 JUNE 86

COMMENT
THIS MODIFIED VERSION DOUBLE PACKS DATA TABLES USING
DOUBLE PRECISION TBLS FOR IBM HARDWARE. NO UNIT
CONVERSION HAS BEEN DONE.

REAL*8 TBLS(1)
INCLUDE 'SESPAR.CMN'
COMMON/S2DIR/LCMX,NRS,LCFW(NREG,NDAT)

C EQUIVALENCE IS USED TO PACK AND UNPACK DATA - GAM
REAL*8 DDATA
REAL*4 SDATA(2)
EQUIVALENCE (DDATA,SDATA(1))
DATA TFAC,RFAC,TCFA/-3.0,0.0,-11.935355/
C . . UNITS...TEMP IN KEV...RHO IN G/CC...TC IN JERK/(KEV*CM*SHAKE)
CALL MATCHK(MID,NRS,LCFW(1,IDT),TBLS(1),IFL)
IF(IFL.EQ.0) GO TO 10
LCFW(IR,IDT)=IFL
IFL=2
RETURN
10 NL=LCMX-LCNT-1
CALL TAbSeq(MID,503.,LU,TBLS(LCNT+2),NL,IFL)
IF(IFL.LE.0) RETURN

C.. CON VerT unITS
NR=TBLS(LCNT+2)
NT=TBLS(LCNT+3)
IPT=LCNT+3+NR+NT

C.. CON VeRT OpACITY tO CO nDUCTIVITY
DO 60 I=1,NT
TBLS(LCNT+3+NR+I)=TBLS(LCNT+3+NR+I)+TFAC
DO 60 J=1, NR
IF(I.GT.1) GO TO 50
TBLS(LCNT+3+J)=TBLS(LCNT+3+J)+RFAC
60 CONTINUE
IPT=IPT+1
TC=TBLS(IPT)

C.. CO nDUCTIVITY IN W/(KELVIN*ME TER)
TC=3.674784D0*(TBLS(3+NR+I+LCNT)-TFAC)+
$ -TC-TBLS(3+LCNT+J)

C.. CON VeRT tO USER UNITS
TC=TC+TCFA
SDATA(1)=TC
SDATA(2)=0.
TBLS(IPT)=DDATA

60 CONTINUE
TBLS(LCNT)=DFLOAT(MID)
TBLS(LCNT+1)=DFLOAT(IDT)
LCFW(IR,IDT)=LCNT
LCNT=LCNT+1+IFL+2
IFL=1
RETURN
END

SUBROUTINE INBUF(LU,Z,N,IM)

C INBUF ReaDS FROm DISC OR TAPE INTO LOCAL ARRAY. IT DOES A C BINARY READ AND TEST FOR END OF FILE. IT ASSUMES THAT THE C DATA RECORDS WERE WRITTEN WITH OUTBUF AND SHOULD BE MADE C COMPATIBLE WITH THAT ROUTINE
C
C ARGUMENTS: LU (INPUT) - 10 UNIT NUMBER OF TAPE, DISC, ETC.
C Z (IN/OUT) - NAME OF LOCAL ARRAY
C N (INPUT) - NUMBER OF WORDS TO BE READ
C IM (OUTPUT) - END FILE FLAG
C = 0, END FILE MARK ENCOUNTERED
C = 1, NO END FILE MARK
C
C MODIFIED BY: G.A. MOSES FOR IMPLEMENTATION ON IBM HARDWARE FOR C INR-KFK MAY 1986
C
C---------------------------------------------------------------
REAL*4 Z(*),
READ(LU,END=1) (Z(i), I=1,N)
GO TO 2
1 IM = 0
RETURN
2 IM = 1
RETURN
END
SUBROUTINE MATCHK(MID,NRS,LOC,TBLs,IFLG)
C----------------------------------------
C MATCHK CHECKS TO SEE IF A MATERIAL HAS BEEN PREVIOUSLY LOADED
C
C ARGUMENTS: MID (INPUT) SESAME MATERIAL ID
C NRS (INPUT) NUMBER OF REGIONS
C LOC (INPUT) ARRAY OF FIRST WORD LOCATIONS
C IN TABLE STORAGE ARRAY FOR
C EACH REGION
C TBLs (INPUT) TABLE STORAGE ARRAY
C IFLG (OUTPUT) =0 MATERIAL NOT PREVIOUSLY LOADED
C GT,0 LOCATION OF TABLE IF LOADED
C ALREADY
C
C MODIFIED BY: G.A. MOSES FOR IMPLEMENTATION ON IBM HARDWARE FOR
C INR-KFK MAY 1986
C----------------------------------------

DIMENSION LOC(1)
REAL*8 TBLs(*)
IFLG = 0
DO 100 J = 1,NRS
LC = LOC(J)
IF(LC .LE. 0) GO TO 100
ITEST = TBLs(LC)
IF(MID .EQ. ITEST) GO TO 200
100 CONTINUE
RETURN
200 IFLG = LC
RETURN
END

SUBROUTINE OUTTAB(LOC,IDT,LR,LOUT)
C----------------------------------------
C OUTTAB OUTPUTS THE DATA IN A SESAME TABLE TO UNIT LOUT IN A FORMAT
C SUITABLE FOR USER INSPECTION
C
C ARGUMENTS: LOC (INPUT) POINTER TO TABLE IN TBLs VECTOR
C IDT (INPUT) =1 P&E, =2 Z, =3 OPACITIES
C LR (INPUT) INDICATOR FOR WHETHER THE
C LEFT (LR=1) OR RIGHT (LR=2) HALF
C OF THE PACKED DATA ELEMENT IS TO
C BE OUTPUTTED
C LOUT (INPUT) LOGICAL UNIT NUMBER FOR OUTPUT
C
C WRITTEN BY: G.A. MOSES FOR IMPLEMENTATION ON IBM HARDWARE
C FOR INR KFK MAY 1986
C----------------------------------------

REAL*8 TBLs
COMMON/SESDAT/ TBLs(1)
C THE EQUIVALENT OF SINGLE AND DOUBLE PRECISION VARIABLES IS USED
C TO PACK AND UNPACK DATA IN TBLs
REAL*4 SDATA(2)
REAL*8 DDATA
EQUIVALENCE (DDATA,SDATA(1))
C
DIMENSION DENS(10), DAT(10)
C
GET THE HEADER INFORMATION
MAT = TBLS(LOC)
NR = TBLS(LOC+2)
NT = TBLS(LOC+3)

C WRITE HEADER INFORMATION
   WRITE(LOUT,3002) MAT, IDT, NR, NT, LOC, LR
C
C WRITE 10 DENSITIES AT A TIME EXCEPT FOR THE LAST FEW AND THEN
C WRITE ONLY AS MANY AS THERE ARE LEFT
   DO 20 I = 1,NR,10
      KMAX = MIN(I+9,NR)
      L = 0
   5   CONTINUE
   LMAX = L

C COLLECT THE NEXT 10 DENSITIES
   DO 5 K = 1,KMAX
      DENS(L+1) = TBLS(LOC+3+I+L)
      IF( IDT .GT. 1 ) DENS(L+1) = 10. ** DENS(L+1)
      L = L + 1
   5 CONTINUE

C WRITE A ROW OF DENSITIES
   WRITE(LOUT,3000) (DENS(L), L = 1,LMAX)
   WRITE(LOUT,3003)
C
C NOW LOOP OVER ALL TEMPERATURES AND WRITE DATA FOR THESE DENSITIES
C AND ALL TEMPERATURES
   DO 10 J = 1,NT

C GET THE TEMPERATURE FOR THIS ROW OF DENSITIES
   TEMP = TBLS(LOC+3+NR+J)
   IF( IDT .GT. 1 ) TEMP = 10. ** TEMP

C GET THE DATA FOR THIS ROW OF DENSITIES
   L = 0
   DO 7 K = 1,KMAX
      DDATA = TBLS(LOC+4+NR+NT*(J-1)*NR+L)
      DAT(L+1) = DDATA(LR)
      IF( IDT . GT. 1 ) DAT(L+1) = 10. ** DAT(L+1)
      L = L + 1
   7 CONTINUE
   LMAX = L

C NOW WRITE THE ROW OF DATA
   WRITE(LOUT,3001) TEMP, (DAT(L), L=1,LMAX)
C
10   CONTINUE
20   CONTINUE
C
RETURN

3000 FORMAT('0/'' TEMP \ DENS ' ,1P10E10.3)
3001 FORMAT(1X,1P1E10.3,2X,10E10.3)
3002 FORMAT('0/20X, ' LONG OUTPUT OF SESAME DATA TABLE//' ' /
   1 20X,'SESAME MATERIAL ID0......................',15/
   2 20X,'DATA TYPE ID=PE,2=2,3=OP................',15/
   3 20X,'NUMBER OF DENSITY POINTS..................',15/
   4 20X,'NUMBER OF TEMPERATURE POINTS..............',15/
   5 20X,'POINTER INTO TBLS VECTOR..................',15/
   6 20X,'(LEFT=1, RIGHT=2 PACKED DATA ELEMENT....',15)
3003 FORMAT( ' ')
END
SUBROUTINE T4DAT

T4DAT SEARCHES AND INTERPOLATES DATA AS A FUNCTION OF REGION, DENSITY, TEMPERATURE, AND FREQUENCY USING PACKED SESAME DATA TABLES. INPUT AND OUTPUT IS THROUGH COMMON BLOCKS. THIS IS NOW USABLE FOR 3-DIMENSIONAL TABLES (SUCH AS MULTI-FREQUENCY OPACITY TABLES 592) AS WELL AS 2-DIMENSIONAL ONES.

THE COMMON BLOCKS RTBLK2 AND SESIN HAVE BEEN EXPANDED AND THE INDEXES IH, IHH, NH, LOCH AND IHLAST ADDED FOR THE NEW VARIABLE. THE T4INTP ROUTINE CALLED MUST BE ONE SIMILARLY REVISED.

ARGUMENTS:

COMMON/SESIN/IR,IDT,AR,AT,IBR,IFL,IH
IR (INPUT) - MATERIAL REGION NUMBER
IDT (INPUT) - DATA TYPE INDICATOR
AR (INPUT) - DENSITY
AT (INPUT) - TEMPERATURE
IBR (INPUT) - SPECIFIES VARIABLES REQUIRED
IBR = 0, BOTH VARIABLES
IBR = 1, FIRST HALF VARIABLE ONLY
IBR = 2, SECOND HALF VARIABLE ONLY
IFL (NOT USED)
IH (INPUT) - FREQUENCY GROUP INDEX
(IH=0 --> SINGLE-GROUP DATA)
COMMON/SESOUT/P(3),E(3)
P,E (OUTPUT) - VARIABLES OF FIRST AND SECOND HALF OF PACKED DATA STRING
P(1),E(1) = VALUE OF THE VARIABLES
P(2),E(2) = DENSITY DERIVATIVES
P(3),E(3) = TEMPERATURE DERIVATIVES

MODIFIED BY: G.A. MOSES FOR IMPLEMENTATION ON IBM HARDWARE FOR INR-KFK MAY 1986

INCLUDE 'SESPAR.CMN'
COMMON/SZDIR/ LCMX,NRS,LCFW(NREG,NDAT)
DIMENSION LOCLST(NDAT),IYLST(NDAT),IYLST(NDAT),IYLST(NDAT),
& IYLST(NDAT),IYLST(NDAT),
COMMONT RTBLK2/LOCX,IX,NX,LOCY,IV,NY,LOCZ,NZ,NSFT,
& R,T,Z(3),INT,IDS,ZZ(IDCIM),IHM,LOCH,NH
REAL*8 TBLS
COMMON/SESOUT/ TBLS(1)
REAL*8 AR, AT
COMMON/SESIN/IR,IDT,AR,AT,IBR,IFL,IH
COMMON/SESOUT/P(3),E(3)
DATA IPLAST/NDAT*1/
DATA IYLST/NDAT*1/
DATA IYLST/NDAT*1/
DATA IYLST/NDAT*1/
DATA LOCLST/NDAT*0/
C LOC IS POINTER TO START OF DATA STRING FOR REGION IR
LOC = LCFW(IR,IDT)+2
N2=1
NX=TBLS(LOC)
NY=TBLS(LOC+1)
IF(IH.EQ.0)GO TO 610
C POINTERS FOR MULTIGROUP DATA
NH=TBLS(LOC+2)
LOCX=LOC+3
LOCY=LOCX+NY
LOCX=LOCY+NX
LOCZ=LOCX+NH
GO TO 620
C POINTERS FOR SINGLE GROUP DATA
610 LOCX=LOCX+2
LOCY=LOCX+NX
LOCZ=LOCY+NY
620 CONTINUE
LOCNX=LOCX+NX-2
LOCNY=LOCX+NY-2
IF(LC1.EQ.LC1LST(IDT)) GO TO 2
LC1LST(IDT)=LOC
IX=NX/2
IX=NY/2
IXLAST(IDT)=0
IYLAST(IDT)=0
GO TO 3
2 IX=IXLAST(IDT)
IX=IXLAST(IDT)
3 R=AR
T=AT
C SEARCH FOR DENSITY INDEX
5 LOC1=LOCX+IX+1
10 IF(R LT. TBL5(LOCI)) GO TO 15
20 IF(LC1.EQ.LC1X+1) GO TO 20
LOC1=LOC1+1
GO TO 10
15 IF(LC1.EQ.LCX) GO TO 20
LOC1=LOC1-1
IF(R LT. TBL5(LOCI)) GO TO 15
20 IX=LOC1-LOCX+1
C SEARCH FOR TEMPERATURE INDEX
LOC1=LOCY+IV-1
30 IF(T LT. TBL5(LOCI)) GO TO 35
40 IF(T LT. TBL5(LOCI+1)) GO TO 40
45 IF(LOC1.EQ.LCNV) GO TO 40
LOC1=LOC1+1
GO TO 30
35 IF(LOC1.EQ.LCX) GO TO 40
LOC1=LOC1-1
IF(T LT. TBL5(LOCI)) GO TO 35
40 IY=LOC1-LOCY+1
C IF ISAME = 0 DENSITY. TEMPERATURE (AND GROUP) INDICES ARE
C THE SAME AS IN THE LAST CALL TO THIS ROUTINE
ISAME = IABS(IX-IXLAST(IDT))+IABS(IY-IYLAST(IDT))
IH=1
IF(IH, EQ, 0) GO TO 850
C MULTI-GROUP ADDITIONS
ISAME=ISAME+IABS(IH-IHLAST(IDT))
IHLAST(IDT)=IH
IH=1
650 IXLAST(IDT) = IX
IYLAST(IDT) = IY
IDS=(IDT-1)*32+1
IPLAST(IDT)=MIND(1, [PLAST(IDT)] + ISAME)
IELAST(IDT)=MIND(1, ELAST(IDT)+ISAME)
IF(1, EQ, 2) GO TO 50
INT=1PLAST(IDT)
NSFT = 1
CALL T4INTP
P(1)=Z(1)
P(2)=Z(2)
P(3)=Z(3)
IPLAST(IDT)=0
IF(IBM.EQ.1) RETURN
INT=IELAST(IDT)
IDS=IDS+16

60 NSFT = 2
CALL T4INTP
E(1)=Z(1)
E(2)=Z(2)
E(3)=Z(3)
IELAST(IDT)=0
RETURN

END

SUBROUTINE T4INTP

THE ROUTINE INTERPOLATES FOR A FUNCTION Z(X,Y) OR Z(X,Y,H) AND ITS
DERIVATIVES FROM TABLES LOCATED IN ARRAY TBL5.

THE ROUTINE REQUIRES COMMON BLOCKS,
COMMON/RTBLK5/LOCX,IX,NX,LOCY,IY,NY,LOCZ,NZ,NSFT,
X,Y,Z(3),IP,IDS,ZZ

LOCX = LOCATION OF X VECTOR
IX = INDEX OF X VECTOR
NX = LENGTH OF X VECTOR
LOCY = LOCATION OF Y VECTOR
IY = INDEX OF Y VECTOR
NY = LENGTH OF Y VECTOR
LOCH = LOCATION OF H VECTOR
IM = INDEX OF H VECTOR
NH = LENGTH OF H VECTOR
LOCZ = LOCATION OF Z(X,Y,H) ARRAY
NZ = SPACING OF Z ARRAY
NSFT = LEFT OR RIGHT HALF PACKED DATA SELECTOR
X,Y (INPUT) - INDEPENDENT VARIABLES
Z (OUTPUT) - VECTOR OF LENGTH 3, WHERE
Z(1) = VALUE OF FUNCTION
Z(2) = X DERIVATIVE OF FUNCTION
Z(3) = Y DERIVATIVE OF FUNCTION
ZZ (IN/OUT) - COEFFICIENT VECTOR OF LENGTH 16
IP (INPUT) - BRANCH PARAMETER
IP.EQ.0, USE INPUT COEFFICIENTS IN ZZ
IP.NE.0, CALCULATE ZZ VECTOR FIRST
IDS (INPUT) - DISPLACEMENT INTO ZZ FOR COEFS.
TO BE USED
COMMON/INTORD/IFN.
IFN (INPUT) - INTERPOLATION TYPE
IFN.NE.1, RATIONAL FUNCTION
IFN.EQ.1, BILINEAR
COMMON/SASOAT/TBL5
TBL5 IS THE TABLE STORAGE ARRAY

REMARKS: UNLESS BILINEAR FORM IS SPECIFIED, ROUTINE
USES RATIONAL FUNCTION METHOD WITH QUADRATIC
ESTIMATE OF DERIVATIVES AT THE MESH POINTS.

INCLUDE 'SESPAR.CMN'
COMMON/RTBLK5/LOCX,IX,NX,LOCY,IY,NY,LOCZ,NZ,NSFT,X,Y,Z(3),
& IP,IDS,ZZ(ICDIM),IH,LOCH,NH
COMMON/INTORD/IFN
REAL*8 TBLS
COMMON/SESDAT/ TBLS(1)
REAL*8 DDATA
REAL*4 SDATA(2)
EQUIVENLANCE (DDATA,SDATA(1))

C
C CALCULATE COEFFICIENTS FOR RATIONAL FUNCTION INTERPOLATION
IF(IFN.EQ.1) GO TO 13
IF(IFN.EQ.0) GO TO 8
I = LOCX+IX-1
IZ = LOCZ+NZ*(IX-1+NX*(IV-1)+NX+NY*(IH-1)
KZ = NZ
IBR = IY
NBR = NY-IX
ZZ(IDS+4) = TBLS(I)
DO 7 K=1,4
K=IDS+K-1
IF(K,LT,4) GO TO 1
IZ = IZ+NZ
GO TO 4
1 IF(K,LT,3) GO TO 2
ZZ(IDS+6) = D
I = LOCY+IV-1
KZ = KZ+NX
IZ = IZ-KZ
IBR = IY
NBR = NY-IV
ZZ(IDS+5) = TBLS(I)
GO TO 3
2 IF(K,LT,2) GO TO 3
IZ = IZ+NX*NZ
GO TO 4
3 D = TBLS(I+1)-TBLS(I)
C4 ZZ(KI)=SHIPT(TBLS(IZ),NSFT)
S=SHIPT(TBLS(IDS),NSFT)
C NOTE CHANGE IN UNPACKING FUNCTION
4 ZZ(KI)=0
C FLD(0,18,ZZ(KI))=FLD(NSFT,18,TBLS(IZ))
DDATA = TBLS(IZ)
ZZ(KI) = SDATA(NSFT)
S=0.
C FLD(0,18,S)=FLD(NSFT,18,TBLS(IZ+KZ))
DDATA = TBLS(IZ+KZ)
S = SDATA(NSFT)
S = (S-ZZ(KI))/D
IF(NBR.EQ.1) GO TO 5
C SP=SHIPT(TBLS(IZ+KZ),NSFT)
C NOTE CHANGE IN UNPACKING FUNCTION
SP=0.
C FLD(0,18,SP)=FLD(NSFT,18,TBLS(IZ+KZ+KZ))
DDATA = TBLS(IZ+KZ+KZ)
SP = SDATA(NSFT)
SP = (SP-D*S-ZZ(KI))/(TBLS(I+2)-TBLS(I+1))
G2 = (SP-S)/(TBLS(I+2)-TBLS(I))
IF(IBR.EQ.1) GO TO 5
IF(S*(S-D*G2).LE.0.) G2=S/D
G1 = G2
G1 = G2
GO TO 6
5 DM = TBLS(I)-TBLS(I-1)
C FLD(0,18,SM)=FLD(NSFT,18,TBL$S(IZ-KZ))
DDATA = TBL$S(IZ-KZ)
SM = SDATA(NSFT)
SM = (ZZ(KI)-SM)/DM
G1 = (S-SM)/(0-DM)
IF(NBR.EQ.1) G2=G1
IF(IBR.GT.2) GO TO 6
IF(SM*(SM-DM*G1).LE.0.) G1=(S-SM-SM)/D
6 IF(G2,NE.0.) G1=G1/G2
ZZ(KI+8) = G1
7 ZZ(KI+12) = G2
ZZ(IDS+7)=D
ZZ8=ZZ(IDS+7)
ZZ7=ZZ(IDS+6)
ZZ(IDS+2)=(ZZ(IDS+1)-ZZ(IDS))/ZZ8
ZZ(IDS+1)=(ZZ(IDS+3)-ZZ(IDS))/ZZ7
ZZ(IDS+3)=(S-ZZ(IDS+2))/ZZ7
ZZ(IDS+12)=ZZ(IDS+12)/ZZ8
ZZ(IDS+13)=ZZ(IDS+13)/ZZ8
ZZ(IDS+14)=ZZ(IDS+14)/ZZ7
ZZ(IDS+15)=ZZ(IDS+15)/ZZ7
C EVALUATE RATIONAL FUNCTION FROM PRECALCULATED COEFFICIENTS
8 QX = X-ZZ(IDS+4)
RX = ZZ(IDS+6)-QX
QV = Y-ZZ(IDS+5)
RV = ZZ(IDS+7)-QV
IF(RX,NE.0.) GO TO 9
W1 = 1.
W2 = 1.
GO TO 10
9 W1 = 1.-1./(1.+ABS(ZZ(IDS+8)*QX/RX))
W2 = 1.-1./(1.+ABS(ZZ(IDS+9)*QX/RX))
10 F1 = ZZ(IDS+12)*W1+ZZ(IDS+8)*(1.-W1))
F2 = ZZ(IDS+13)*W2+ZZ(IDS+9)*(1.-W2))
Z(2) = ZZ(IDS+6)*(RY*(F1-ZZ(IDS+12))*W1+QY*(F2-ZZ(IDS+13))*W2)
G1 = RV*F1+QV*F2
IF(RV,NE.0.) GO TO 11
W1 = 1.
W2 = 1.
GO TO 12
11 W1 = 1.-1./(1.+ABS(ZZ(IDS+10)*QV/RV))
W2 = 1.-1./(1.+ABS(ZZ(IDS+11)*QV/RV))
12 F3 = ZZ(IDS+14)*W1+ZZ(IDS+10)*(1.-W1))
F4 = ZZ(IDS+15)*W2+ZZ(IDS+11)*(1.-W2))
Z(3) = ZZ(IDS+7)*(RX*(F3-ZZ(IDS+14))*W1+QX*(F4-ZZ(IDS+15))*W2)
G2 = RX*F3+QX*F4
ZZ2=ZZ(IDS+1)
ZZ3=ZZ(IDS+2)
ZZ4=ZZ(IDS+3)
Z(1) = ZZ(IDS)+ZZ2+ZZ4*QY-RX*G1)*QX+(ZZ3-RY*G2)*QY
Z(2) = Z(2)+ZZ2+QY*(ZZ4+RY*(F3-F4))+(QX-RX)*G1
Z(3) = Z(3)+ZZ3+QX*(ZZ4+RX*(F1-F2))+(QY-RY)*G2
RETURN
C CALCULATE COEFFICIENTS FOR BILINEAR INTERPOLATION
13 IF(IP.EQ.0) GO TO 14
I=LOCX+IX
IND=IDS+4
ZZ(IND)=TBL$S(I-1)
DX=TBL$S(I-1)-ZZ(IND)
J=LOCY+IY
IND=IDS+5
ZZ(IND)=TBLS(J-1)
DV=TBLS(J)-ZZ(IND)
IZ=LOCZ+NZ*(IX-1+NX*(IV-1))
C ZZ(IND)=SHIFT(TBLS(I),NSFT)
C NOTE CHANGE IN UNPACKING FUNCTION
ZZ(IDS)=0.
C FLD(0,18,ZZ(IDS))=FLD(NSFT,18,TBLS(IZ))
DDATA = TBLS(I)
ZZ(IDS) = SDATA(NSFT)
IND=IDS+1
C ZZ(IND)=SHIFT(TBLS(IZ+NZ),NSFT)
C NOTE CHANGE IN UNPACKING FUNCTION
ZZ(IND)=0.
C FLD(0,18,ZZ(IND))=FLD(NSFT,18,TBLS(IZ+NZ))
DDATA = TBLS(IZ+NZ)
ZZ(IND) = SDATA(NSFT)
IZ(IDS)=(ZZ(IND)-ZZ(IDS))/DX
IZ=IZ+NZ*NX
IND=IDS+2
C ZZ(IND)=SHIFT(TBLS(IZ),NSFT)
C NOTE CHANGE IN UNPACKING FUNCTION
ZZ(IND)=0.
C FLD(0,18,ZZ(IND))=FLD(NSFT,18,TBLS(IZ))
DDATA = TBLS(I)
ZZ(IND) = SDATA(NSFT)
IZ(IND)=(ZZ(IND)-ZZ(IDS))/DY
IND=IDS+3
C ZZ(IND)=SHIFT(TBLS(IZ+NZ),NSFT)
C NOTE CHANGE IN UNPACKING FUNCTION
ZZ(IND)=0.
C FLD(0,18,ZZ(IND))=FLD(NSFT,18,TBLS(IZ+NZ))
DDATA = TBLS(IZ+NZ)
ZZ(IND) = SDATA(NSFT)
IZ(IND)=(ZZ(IND)-ZZ(IDS)-ZZ(IDS+1)*DX-IZ(IDS+2)*DY)/(DX*DY)
C EVALUATE BILINEAR FUNCTION FROM PRECALCULATED COEFFICIENTS
QX = X-IZ(IDS+4)
QY = Y-IZ(IDS+5)
Z(2) = ZZ(IDS+1)+ZZ(IDS+3)*QV
Z(3) = ZZ(IDS+2)+ZZ(IDS+3)*QX
Z(1) = ZZ(IDS)+Z(2)*QX+ZZ(IDS+2)*QY
RETURN
END

SUBROUTINE TABSEQ(MID,TID,LIB,AD,LEN,IFLAG)
C TABSEQ FETCHES TABLES FROM A SESAME LIBRARY USING SEQUENTIAL I/O

C ARGUMENTS
C MID; (INPUT) SESAME MATERIAL ID
C TID; (INPUT) TABLE NO. - IF 0,0 MATERIAL
C LIB; (INPUT) LIBRARY FILE I/O UNIT NO.
C AD; (OUTPUT) REAL*8 ARRAY FOR TABLE STORAGE
C LEN; (INPUT) LENGTH OF A
C IFLAG; (OUTPUT) FLAG
C =0 DATA NOT FOUND
C GT.0 NO OF WORDS IN TABLE RETURNED
C LT. 0 - NO. EXTRA WORDS NEEDED

C MODIFIED BY: G.A. MOSES FOR IMPLEMENTATION ON IBM HARDWARE FOR
C INR-KFK MAY 1986
DIMENSION HINDEX(35)
REAL*8 AD(*)

C BUFFER ARRAY TO READ SINGLE PRECISION TABLE. AT THE END OF TABSEQ
C THE DATA IS TRANSFERRED TO THE DOUBLE PRECISION AD ARRAY SO THAT IT
C CAN BE DOUBLE PACKED IN THE GET*** ROUTINES THAT CALL TABSEQ.
C
REAL*4 A(6000)

DATA LBLST/0/
DATA HINDEX(1)/0.0/
IFLAG = 0
IF(LIB .NE. LBLST) GO TO 50
IDLAST = HINDEX(I)
IF(IDLAST .NE. MID) GO TO 50
IF(TID .NE. 0.0) GO TO 230
NW = HINDEX(5)
NW = NW + NW + 5
IF(LIB .LT. NW) GO TO 999
DO 30 J = 1,NW
A(J) = HINDEX(J)
30 CONTINUE
IFLAG = NW
RETURN

50 LBLST = LIB
NW = 1
IF(LIB .LT. NW) GO TO 999
REWRIND LIB
CALL INBUF(LIB,A,NW,IM)
N = A(1)
NW = N + N
IF(LIB .LT. NW) GO TO 999
CALL INBUF(LIB,A,NW,IM)
IX = 0
DO 100 J = 1,N
IX = IX + 1
ITEST = A(J)
IF(ITEST .EQ. MID) GO TO 200
100 CONTINUE
RETURN

200 NW = A(IX+N)
IF(LIB .LT. NW) GO TO 999
NX = 0
205 CALL INBUF(LIB,A,1,IM)
IF(IM .NE. 0) GO TO 205
NX = NX + 1
IF(NX .EQ. IX) GO TO 208
GO TO 205
208 CALL INBUF(LIB,A,NW,IM)
DO 210 J=1,NW
HINDEX(J) = A(J)
210 CONTINUE
IF(TID .EQ. 0.0) GO TO 500

IX = 0
N = HINDEX(5)
DO 300 J=1,N
IX = IX + 1
IF(TID .EQ. HINDEX(5+IX)) GO TO 400
300 CONTINUE
RETURN

400 NW = HINDEX(5+IX+N)
IF(LEN .LT. NW) GO TO 999
DO 420 J=1,IX
IF(J .LT. IX) GO TO 410
CALL INBUF(LIB,A,NW,IM)
GO TO 420
410 CALL INBUF(LIB,A,1,IM)
420 CONTINUE
DO 440 J=1,IX
BACKSPACE LIB
440 CONTINUE
500 IFLAG = NW
C
C NOW WE MUST PUT THE SINGLE PRECISION DATA INTO THE DOUBLE PRECISION
C TABLE SO THAT IT WILL PACK
DO 600 J = 1,NW
AD(J) = A(J)
600 CONTINUE
RETURN
999 IFLAG = LEN - NW
RETURN
END
Dear Colleague:

Since October 1984, we have been the distributor of the Los Alamos National Laboratory's T-4 Material Properties Data Library (Sesame). This Library contains equation of state (EOS), opacity, and conductivity data for about 90 different materials in each of these three data categories. At this time electrical and thermal conductivities are available only for pure elements, but the EOS and opacities are also available for more complex materials.

Tables typically cover the density range from $10^{-4}$ to $10^{4}$ of normal density, and temperatures from about 0 to 10 keV ($1$ eV = $11605$ K) for EOS, $1$ eV to $100$ keV for opacities if materials contain no element heavier than $Z = 30$, and $50$ eV to $100$ keV for all other materials. The three types of tables provide data for all three modes of energy transport (convection, radiation, and conduction) over wide ranges of engineering and science applications.

Each material has a number of sub-tables which include comments, basic data, total EOS, ion EOS, electron EOS, Rosseland opacity, Planck opacity, mean ion charge, electrical conductivity, thermal conductivity, thermoelectric coefficient, electron conductive opacity, etc. The attached list summarizes these sub-tables. Some of these sub-tables permit two-temperature applications of the EOS, e.g., the ion temperature may be different than the electron temperature. Other tables appear to be duplicates, e.g., the mean ion charge, or the electron conductive opacity: actually they are quantities based on different atomic models.

A list of the tables that are currently available is attached. Table numbers with four digits correspond to EOS, five digits starting with a 1 correspond to opacities, and starting with a 2 to conductivities. Each material has a 3 digit material index which appears in the thousands, hundreds, and tens positions of the material table number. The unit position of the material table number indicates the version number of that table. Different version numbers usually indicate a different density-temperature grid, but can also indicate other differences such as the inclusion of the vapor dome in an EOS or a Maxwell construct in its place, or the inclusion of certain trace elements that affect the opacity but not the EOS or the conductivity. These differences are summarized in the comments table of the above mentioned sub-tables that are included in each material table. New materials are constantly added, others are replaced with improved versions.

The tables are available from us on a standard 9-track mainframe computer tape, at 1600 bpi, blocked 100 card images per record, 80 characters per card image, in either EBCDIC or ASCII format. The user should specify EBCDIC or ASCII when ordering. Other than 100 card blocking is also possible, but much shorter blocks may require more than one tape reel. There are about 28,000,000 characters of information. A subroutine package (SesPack) is also available. Some of these subroutines may be useful for inclusion into the user's computer program to facilitate accessing the Library and for interpolation. All documentation is contained in the Library, but some additional printed documentation is also available.

A fee schedule is enclosed. Costs include the tape, packing, standard air mail postage, and export license for foreign users. We are looking forward to serving you!

Sincerely yours,

W. F. Huebner

W. F. Huebner
LOS ALAMOS SCIENCE LIAISON COMPANY
A Science, Technology, and Information Transfer Company
253 Rio Bravo
Los Alamos, NM 87544
Tel.: (505) 672-3301
1 October, 1984

FEES SCHEDULE FOR EQUATION OF STATE LIBRARY (SESAME)

A. Computer Programs for use of Library  $100.--
B. Equation of State Data  100.--
C. Opacity Data  100.--
D. Conductivity Data  100.--

There is a discount of $50.-- when any two data categories (B, C, or D) are ordered, and $100.-- if all three data categories are ordered.

Updates of data categories B, C, or D are half price if the order is placed within 12 months of the original order or the last update order.

An additional $50.-- is required for shipments outside the USA, Canada, or Mexico.

Payment is in US dollars, payable to LOS ALAMOS SCIENCE LIAISON COMPANY and must accompany your order.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>126</td>
<td>Actinium</td>
<td>7410</td>
<td>17410</td>
<td></td>
</tr>
<tr>
<td>371</td>
<td>Alumina</td>
<td>3715</td>
<td></td>
<td></td>
</tr>
<tr>
<td>371</td>
<td>Aluminum</td>
<td>3711</td>
<td>13710</td>
<td>23713</td>
</tr>
<tr>
<td>371</td>
<td>Aluminum</td>
<td>3716</td>
<td></td>
<td></td>
</tr>
<tr>
<td>371</td>
<td>Aluminum</td>
<td>3712</td>
<td></td>
<td></td>
</tr>
<tr>
<td>371</td>
<td>Aluminum</td>
<td>3713</td>
<td></td>
<td></td>
</tr>
<tr>
<td>190</td>
<td>Americium</td>
<td>5520</td>
<td></td>
<td></td>
</tr>
<tr>
<td>288</td>
<td>Ammonia</td>
<td>5171</td>
<td>15170</td>
<td>25171</td>
</tr>
<tr>
<td>517</td>
<td>Argon</td>
<td>5172</td>
<td></td>
<td></td>
</tr>
<tr>
<td>227</td>
<td>Arsenic</td>
<td>227</td>
<td></td>
<td></td>
</tr>
<tr>
<td>144</td>
<td>Astadine</td>
<td>144</td>
<td></td>
<td></td>
</tr>
<tr>
<td>208</td>
<td>Barium</td>
<td>208</td>
<td></td>
<td></td>
</tr>
<tr>
<td>202</td>
<td>Beryllium</td>
<td>202</td>
<td></td>
<td></td>
</tr>
<tr>
<td>202</td>
<td>Beryllium Oxide</td>
<td>202</td>
<td></td>
<td></td>
</tr>
<tr>
<td>322</td>
<td>Bismuth</td>
<td>322</td>
<td></td>
<td></td>
</tr>
<tr>
<td>233</td>
<td>Boron</td>
<td>233</td>
<td></td>
<td></td>
</tr>
<tr>
<td>708</td>
<td>Boron Carbide</td>
<td>708</td>
<td></td>
<td></td>
</tr>
<tr>
<td>410</td>
<td>Brass</td>
<td>410</td>
<td></td>
<td></td>
</tr>
<tr>
<td>505</td>
<td>Bromine</td>
<td>505</td>
<td></td>
<td></td>
</tr>
<tr>
<td>553</td>
<td>Butane (normal)</td>
<td>553</td>
<td></td>
<td></td>
</tr>
<tr>
<td>266</td>
<td>Cadmium</td>
<td>266</td>
<td></td>
<td></td>
</tr>
<tr>
<td>203</td>
<td>Calcium</td>
<td>203</td>
<td></td>
<td></td>
</tr>
<tr>
<td>733</td>
<td>Calcium Carbonate</td>
<td>733</td>
<td></td>
<td></td>
</tr>
<tr>
<td>783</td>
<td>Carbon Liquid</td>
<td>783</td>
<td></td>
<td></td>
</tr>
<tr>
<td>794</td>
<td>Carbon Phenolic</td>
<td>794</td>
<td></td>
<td></td>
</tr>
<tr>
<td>720</td>
<td>Carbon Tetrafluoride</td>
<td>720</td>
<td></td>
<td></td>
</tr>
<tr>
<td>317</td>
<td>Cerium</td>
<td>317</td>
<td></td>
<td></td>
</tr>
<tr>
<td>851</td>
<td>Cesium</td>
<td>851</td>
<td></td>
<td></td>
</tr>
<tr>
<td>502</td>
<td>Chlorine</td>
<td>502</td>
<td></td>
<td></td>
</tr>
<tr>
<td>588</td>
<td>Chlorine Oxide</td>
<td>588</td>
<td></td>
<td></td>
</tr>
<tr>
<td>307</td>
<td>Chromium</td>
<td>307</td>
<td></td>
<td></td>
</tr>
<tr>
<td>312</td>
<td>Cobalt</td>
<td>312</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9333</td>
<td>Copper</td>
<td>9333</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9333</td>
<td>Copper</td>
<td>9333</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9333</td>
<td>Copper</td>
<td>9333</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9333</td>
<td>Copper</td>
<td>9333</td>
<td></td>
<td></td>
</tr>
<tr>
<td>182</td>
<td>Curium</td>
<td>182</td>
<td></td>
<td></td>
</tr>
<tr>
<td>799</td>
<td>DAP (Diallyl Phthalate)</td>
<td>799</td>
<td></td>
<td></td>
</tr>
<tr>
<td>526</td>
<td>Deuterium</td>
<td>526</td>
<td></td>
<td></td>
</tr>
<tr>
<td>527</td>
<td>Deuterium-Tritium</td>
<td>527</td>
<td></td>
<td></td>
</tr>
<tr>
<td>783</td>
<td>Diamond</td>
<td>783</td>
<td></td>
<td></td>
</tr>
<tr>
<td>503</td>
<td>Dry Air</td>
<td>503</td>
<td></td>
<td></td>
</tr>
<tr>
<td>212</td>
<td>Dysprosium</td>
<td>212</td>
<td></td>
<td></td>
</tr>
<tr>
<td>760</td>
<td>Epoxy (Epon 828)</td>
<td>760</td>
<td></td>
<td></td>
</tr>
<tr>
<td>760</td>
<td>Epoxy (Epon 828)</td>
<td>760</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2285</td>
<td>Erbium</td>
<td>2285</td>
<td></td>
<td></td>
</tr>
<tr>
<td>761</td>
<td>Europium</td>
<td>761</td>
<td></td>
<td></td>
</tr>
<tr>
<td>765</td>
<td>Fiberglass</td>
<td>765</td>
<td></td>
<td></td>
</tr>
<tr>
<td>504</td>
<td>Fluorine</td>
<td>504</td>
<td></td>
<td></td>
</tr>
<tr>
<td>691</td>
<td>Formaldehyde</td>
<td>691</td>
<td></td>
<td></td>
</tr>
<tr>
<td>134</td>
<td>Francium</td>
<td>134</td>
<td></td>
<td></td>
</tr>
<tr>
<td>320</td>
<td>Gadolinium</td>
<td>320</td>
<td></td>
<td></td>
</tr>
<tr>
<td>397</td>
<td>Gallium</td>
<td>397</td>
<td></td>
<td></td>
</tr>
<tr>
<td>395</td>
<td>Germanium</td>
<td>395</td>
<td></td>
<td></td>
</tr>
<tr>
<td>270</td>
<td>Gold</td>
<td>270</td>
<td></td>
<td></td>
</tr>
<tr>
<td>389</td>
<td>Hafnium</td>
<td>389</td>
<td></td>
<td></td>
</tr>
<tr>
<td>576</td>
<td>Helium</td>
<td>576</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

C-3
<table>
<thead>
<tr>
<th>#</th>
<th>Element</th>
<th>5760</th>
<th>15760</th>
<th>25761</th>
</tr>
</thead>
<tbody>
<tr>
<td>526</td>
<td>Helium</td>
<td>5761</td>
<td>16210</td>
<td></td>
</tr>
<tr>
<td>821</td>
<td>HE-9404</td>
<td>3660</td>
<td>23661</td>
<td></td>
</tr>
<tr>
<td>364</td>
<td>Holmium</td>
<td>5251</td>
<td></td>
<td></td>
</tr>
<tr>
<td>525</td>
<td>Hydrogen</td>
<td>5250</td>
<td>15250</td>
<td>25251</td>
</tr>
<tr>
<td>321</td>
<td>Indium</td>
<td>15070</td>
<td>25071</td>
<td></td>
</tr>
<tr>
<td>507</td>
<td>Iodine</td>
<td>23741</td>
<td></td>
<td></td>
</tr>
<tr>
<td>374</td>
<td>Iridium</td>
<td>2145</td>
<td></td>
<td></td>
</tr>
<tr>
<td>224</td>
<td>Iron</td>
<td>2140</td>
<td>12140</td>
<td>22143</td>
</tr>
<tr>
<td>584</td>
<td>Isobutane</td>
<td>6540</td>
<td></td>
<td></td>
</tr>
<tr>
<td>518</td>
<td>Krypton</td>
<td>5181</td>
<td></td>
<td></td>
</tr>
<tr>
<td>518</td>
<td>Krypton</td>
<td>5180</td>
<td>15180</td>
<td>25161</td>
</tr>
<tr>
<td>386</td>
<td>Lanthanum</td>
<td>23861</td>
<td></td>
<td></td>
</tr>
<tr>
<td>320</td>
<td>Lead</td>
<td>23201</td>
<td></td>
<td></td>
</tr>
<tr>
<td>725</td>
<td>Lithia-Boria Glass</td>
<td>7292</td>
<td>17295</td>
<td></td>
</tr>
<tr>
<td>229</td>
<td>Lithium</td>
<td>2290</td>
<td>12290</td>
<td>22291</td>
</tr>
<tr>
<td>229</td>
<td>Lithium</td>
<td>2291</td>
<td></td>
<td></td>
</tr>
<tr>
<td>724</td>
<td>Lithium Deuteride</td>
<td>7243</td>
<td>17241</td>
<td></td>
</tr>
<tr>
<td>724</td>
<td>Lithium Deuteride</td>
<td>7241</td>
<td>17240</td>
<td></td>
</tr>
<tr>
<td>735</td>
<td>Lithium Fluoride</td>
<td>17350</td>
<td></td>
<td></td>
</tr>
<tr>
<td>737</td>
<td>Lithium Hydride</td>
<td>7371</td>
<td>17370</td>
<td></td>
</tr>
<tr>
<td>269</td>
<td>Lutetium</td>
<td></td>
<td></td>
<td>22891</td>
</tr>
<tr>
<td>286</td>
<td>Magnesium</td>
<td>12860</td>
<td>22861</td>
<td></td>
</tr>
<tr>
<td>308</td>
<td>Manganese</td>
<td>13080</td>
<td>23081</td>
<td></td>
</tr>
<tr>
<td>332</td>
<td>Mercury</td>
<td>23321</td>
<td></td>
<td></td>
</tr>
<tr>
<td>650</td>
<td>Methane</td>
<td>5502</td>
<td></td>
<td></td>
</tr>
<tr>
<td>650</td>
<td>Methane</td>
<td>5500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>650</td>
<td>Methane</td>
<td>5501</td>
<td></td>
<td></td>
</tr>
<tr>
<td>752</td>
<td>Mica</td>
<td>7520</td>
<td></td>
<td></td>
</tr>
<tr>
<td>298</td>
<td>Molybdenum</td>
<td>2982</td>
<td></td>
<td></td>
</tr>
<tr>
<td>298</td>
<td>Molybdenum</td>
<td>2981</td>
<td></td>
<td></td>
</tr>
<tr>
<td>298</td>
<td>Molybdenum</td>
<td>2980</td>
<td>12980</td>
<td>22981</td>
</tr>
<tr>
<td>298</td>
<td>Molybdenum</td>
<td>2983</td>
<td></td>
<td></td>
</tr>
<tr>
<td>755</td>
<td>Molar</td>
<td></td>
<td>17550</td>
<td></td>
</tr>
<tr>
<td>311</td>
<td>Neodymium</td>
<td></td>
<td></td>
<td>23111</td>
</tr>
<tr>
<td>541</td>
<td>Neon</td>
<td>5411</td>
<td></td>
<td></td>
</tr>
<tr>
<td>541</td>
<td>Neon</td>
<td>5410</td>
<td>15410</td>
<td>25411</td>
</tr>
<tr>
<td>164</td>
<td>Neptunium</td>
<td></td>
<td></td>
<td>21641</td>
</tr>
<tr>
<td>711</td>
<td>Nevada Alluvium</td>
<td>7111</td>
<td>17111</td>
<td></td>
</tr>
<tr>
<td>310</td>
<td>Nickel</td>
<td>3100</td>
<td>13100</td>
<td>23101</td>
</tr>
<tr>
<td>274</td>
<td>Niobium</td>
<td>12740</td>
<td>22741</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>Nitrogen</td>
<td>5000</td>
<td>15000</td>
<td>25001</td>
</tr>
<tr>
<td>500</td>
<td>Nitrogen</td>
<td>5001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>Osmium</td>
<td>5011</td>
<td></td>
<td></td>
</tr>
<tr>
<td>501</td>
<td>Oxygen</td>
<td>5010</td>
<td>15010</td>
<td>25011</td>
</tr>
<tr>
<td>501</td>
<td>Oxygen</td>
<td></td>
<td>23831</td>
<td></td>
</tr>
<tr>
<td>383</td>
<td>Palladium</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>762</td>
<td>Parylene (Poly(chloropara-xylylene)</td>
<td>17620</td>
<td></td>
<td></td>
</tr>
<tr>
<td>822</td>
<td>PBX-9501 (High Explosive)</td>
<td>16220</td>
<td></td>
<td></td>
</tr>
<tr>
<td>820</td>
<td>PBX-9502 (High Explosive)</td>
<td>8200</td>
<td>18201</td>
<td></td>
</tr>
<tr>
<td>722</td>
<td>Phosphorous Trifluoride</td>
<td>17220</td>
<td></td>
<td></td>
</tr>
<tr>
<td>391</td>
<td>Phosphorus</td>
<td>13910</td>
<td>23911</td>
<td></td>
</tr>
<tr>
<td>373</td>
<td>Platinum</td>
<td>3730</td>
<td></td>
<td></td>
</tr>
<tr>
<td>119</td>
<td>Plutonium</td>
<td>23731</td>
<td></td>
<td></td>
</tr>
<tr>
<td>191</td>
<td>Polonium</td>
<td>21191</td>
<td></td>
<td></td>
</tr>
<tr>
<td>219</td>
<td>Polonium</td>
<td>21911</td>
<td></td>
<td></td>
</tr>
<tr>
<td>738</td>
<td>Polycrystal Quartz</td>
<td>7363</td>
<td></td>
<td></td>
</tr>
<tr>
<td>717</td>
<td>Polyethylene (branched)</td>
<td>7171</td>
<td>17170</td>
<td></td>
</tr>
<tr>
<td>716</td>
<td>Polyethylene (branched, fully deuterated)</td>
<td>7160</td>
<td></td>
<td></td>
</tr>
<tr>
<td>716</td>
<td>Polyethylene (branched, fully deuterated)</td>
<td>7180</td>
<td></td>
<td></td>
</tr>
<tr>
<td>718</td>
<td>Polyethylene (Marlex)</td>
<td>7180</td>
<td></td>
<td></td>
</tr>
<tr>
<td>759</td>
<td>Polystyrene</td>
<td>7590</td>
<td>17590</td>
<td></td>
</tr>
<tr>
<td>Element</td>
<td>Atomic Number</td>
<td>Mass Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------</td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polystyrene Foam</td>
<td>17591</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polystyrene + Krypton</td>
<td>17592</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polytetraethylene (linear)</td>
<td>7230</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polyurethane</td>
<td>7560</td>
<td>17560</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Potassium</td>
<td>12460</td>
<td>22461</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prasodymium</td>
<td>12100</td>
<td>22951</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Promethium</td>
<td></td>
<td>21381</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Protactinium</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quartz</td>
<td>7380</td>
<td>17380</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radium</td>
<td>1680</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radon</td>
<td>25061</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rhenium</td>
<td>23041</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rhodium</td>
<td>23051</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rubidium</td>
<td>22811</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ruthenium</td>
<td>22751</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Salt (Sodium Chloride)</td>
<td>7281</td>
<td>17280</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Samarium</td>
<td>394</td>
<td>23941</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Selenium</td>
<td>301</td>
<td>23011</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Samarium</td>
<td>381</td>
<td>23811</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Silicon</td>
<td>7382</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Silver</td>
<td>7272</td>
<td>22721</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sodium</td>
<td>2448</td>
<td>22441</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Solar Mixture (Ross-Alier)</td>
<td>5280</td>
<td>15280</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spinel</td>
<td>4270</td>
<td>14270</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stainless Steel (304)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strontium</td>
<td>360</td>
<td>23601</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sulfur Dioxide</td>
<td>561</td>
<td>15810</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sulfur Fluoride</td>
<td>563</td>
<td>15830</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sulphur</td>
<td>401</td>
<td>24011</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sylgard (Silicon Rubber)</td>
<td>7932</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tantalum</td>
<td>3522</td>
<td>23521</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Technetium</td>
<td>17190</td>
<td>17190</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Teflon (Polytetrafluoroethylene)</td>
<td>2297</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tellurium</td>
<td>7190</td>
<td>22971</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Terbium</td>
<td>2222</td>
<td>22221</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thallium</td>
<td>3370</td>
<td>223701</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thorium</td>
<td>141</td>
<td>21411</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thulium</td>
<td>338</td>
<td>23381</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tin</td>
<td>216</td>
<td>22161</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Titanium</td>
<td>2592</td>
<td>22961</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tuff</td>
<td>21720</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tungsten</td>
<td>3541</td>
<td>23541</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tungsten Carbide</td>
<td>3560</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uranium</td>
<td>154</td>
<td>21541</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uranium Dioxide</td>
<td>7432</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Urethane</td>
<td>753</td>
<td>17530</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vanadium</td>
<td>255</td>
<td>22551</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Water</td>
<td>7150</td>
<td>22551</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Water</td>
<td>7192</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Westerly Granite</td>
<td>7390</td>
<td>17390</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Xenon</td>
<td>519</td>
<td>25191</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ytterbium</td>
<td>23441</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yttrium</td>
<td>23361</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zink</td>
<td>314</td>
<td>23141</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zirconium</td>
<td>318</td>
<td>23181</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### SESAME TABLES

#### Subtables

- **Tables 101-199**
  - Comments
- **Table 201**
  - Basic Data
- **Table 301**
  - Total EOS
- **Table 303**
  - Ion EOS Plus Cold Curve
- **Table 304**
  - Electron EOS
- **Table 305**
  - Ion EOS (Including Zero Point)
- **Table 306**
  - Cold Curve (No Zero Point)
- **Table 401**
  - Vaporization
- **Table 501**
  - Opacity Grid Boundary: Calculated vs. Interpolated
- **Table 502**
  - Rosseland Mean Opacity (cm² g⁻¹)
- **Table 503**
  - Electron Conductive Opacity¹ (cm² g⁻¹)
- **Table 504**
  - Mean Ion Charge¹ (free electrons per atom)
- **Table 505**
  - Planck Mean Opacity (cm² g⁻¹)
- **Table 601**
  - Mean Ion Charge² (free electrons per atom)
- **Table 602**
  - Electrical Conductivity (sec⁻¹)
- **Table 603**
  - Thermal Conductivity (cm⁻¹ sec⁻¹)
- **Table 604**
  - Thermoelectric Coefficient (cm⁻¹ sec⁻¹)
- **Table 605**
  - Electron Conductive Opacity² (cm² g⁻¹)

¹Opacity Model (Hubbard-Lampe)

²Conductivity Model (Ziman)
Recent Developments in the
Sesame Equation-of-State Library

B. I. Bennett
J. D. Johnson
G. I. Kerley
G. T. Rood
RECENT DEVELOPMENTS IN THE
SESAME EQUATION-OF-STATE LIBRARY

by

B. I. Bennett, J. D. Johnson, G. I. Kerley, and G. T. Rood

ABSTRACT

The Sesame library is a computer file of tabular equations of state and other material properties, developed in Group T-4 of the Los Alamos Scientific Laboratory. This report describes some of the theoretical models used in the library, the structure of the data storage and acquisition system, and the computer routines that have been developed to access and use the tables in practical applications. It also discusses how non-LASL users can obtain the Sesame data and subroutine libraries.

I. INTRODUCTION

The Sesame library is a file of equation-of-state (EOS) tables and other material properties that can be accessed by computer programs for a variety of applications. It has been in use at the Los Alamos Scientific Laboratory (LASL) for several years. Now LASL offers this library, together with interpolation programs and utility routines for manipulating the data files, to other laboratories and research groups. In this report, we will give an overview of the Sesame library and discuss some new developments that should be of interest to users.

During the past 30 years, remarkable achievements have been made in numerical hydrodynamics calculations. Modern computer codes, which include the effects of material strength, fracture, viscosity, chemical reactions, heat conduction, and radiation transport, can be used to make realistic predictions for complicated hydrodynamic flow problems. These numerical codes have become essential tools to workers in many fields, including weapons effects, reactor safety analysis, and laser fusion.
To make accurate hydrodynamics calculations, it is necessary to have realistic EOS and other properties. However, the study of an EOS is a big project in itself. Even when data are readily available, a researcher may have difficulty finding a way to represent them in his code.

The Sesame library was developed to store the best EOS data in a tabular form, which is useful for hydrodynamics codes and other applications. A key feature of the library is that the codes can access the tables directly, instead of using approximate analytic fits to the data. Although analytic formulas can be useful, they cannot represent a complicated EOS table accurately over a wide range of temperatures and densities. We have designed computer subroutines that search the tabular library for a list of materials and calculate the thermodynamic functions by interpolation from the tables. These routines are modular and are written in standard FORTRAN, so that they can be incorporated into user codes with relative ease. Using these techniques, we have found the tabular EOS formulation to be very feasible and useful.

In Sec. II, we discuss some of the theoretical models used in generating Sesame EOS tables. The structure of the data storage and acquisition system is presented in Sec. III. In Sec. IV we describe the subroutine library that has been written for users to access the data for applications.

To obtain additional information about the Sesame library or to be placed on the mailing list for documents, write to

Sesame Library, MS-925
Los Alamos Scientific Laboratory
Los Alamos, NM. 87545.

To obtain Sesame tables, a user should mail his own magnetic tapes and a list of the materials required to the above address. One tape will be written with a copy of the Sesame FORTRAN subroutine library, and the other tapes will be written with tabular data in a BCD card image format. One tape can hold data for up to 25 materials. A set of instructions will be enclosed when the tapes are returned.

II. MODELS FOR CALCULATING EOS

There exists no universal method for calculating the EOS that is valid for all pressures, temperatures, and chemical compositions. Many models can be used,
and studies should be conducted to determine which approach is best for a particular material. It is our goal that the Sesame library should provide the best theoretical and experimental data available. Therefore, the EOS tables can come from any source, including laboratories other than LASL.

In this section, we will discuss some of the models that have been developed at LASL for generating Sesame tables and give a general idea of the concepts involved. It is rarely possible to apply these methods to a specific problem in a straightforward way. For an example of a problem in which more theoretical work is required, the reader should refer to our study of the deuterium EOS.\textsuperscript{2,3}

It is common to write the EOS as a sum of three contributions.\textsuperscript{4} The pressure is given by

\[ P(\rho, T) = P_c(\rho) + P_N(\rho, T) + P_e(\rho, T), \]  \hspace{1cm} (1)

where \( \rho \) is the density and \( T \) is the temperature. \( P_c(\rho) \) is the cold curve, or contribution from electronic cohesive and repulsive forces at \( T = 0 \) K. \( P_N(\rho, T) \) is the contribution from nuclear motion, and \( P_e(\rho, T) \) is the thermal electronic term. Expressions for the internal energy, the Helmholtz free energy, and the other thermodynamic functions are similar to Eq. (1).

A. The Cold Curve

For most practical applications, the cold curve must be determined by empirical methods and phenomenological theories. However, prospects for improving this situation are good. Some calculations for compressed solids have been made using band-theoretical methods,\textsuperscript{5-7} although these computations are too difficult and expensive to perform on a routine basis. Recently, Liberman has developed a relativistic and quantum-mechanical compressed atom model that may be useful for many applications.\textsuperscript{8} Sometimes these more sophisticated techniques can be used to check and improve upon the empirical methods discussed below.

The Thomas-Fermi-Dirac (TFD) theory\textsuperscript{9} and other statistical theories of the atom give reasonable results for the EOS at high densities. Many methods for calculating the cold curve are based upon empirical corrections to the TFD results.

Barnes\textsuperscript{10} has found that the cold curve for many materials can be described by
\[ P_C(\rho) = \frac{2}{3} \left( n e^{-\rho} - e^{-\rho} \right). \tag{2} \]

where \( n = \rho/\rho_0 \), \( \nu = 1 - n^{-1/3} \), and \( \rho_0 \) is the normal density of the solid. The constants \( a \) and \( b_r \) are obtained by fitting the TFD data at high densities, and \( b_a \) is determined from the experimental bulk modulus, \( B_0 \), using the relation

\[ B_0 = \left( \frac{\rho d P_C}{d \rho} \right)_{\rho_0} = \frac{1}{3} a \left( 3 + b_r - b_a \right). \tag{3} \]

Equation (2) has been used in generating many of the Sesame EOS tables. For materials having phase transitions that involve large volume changes, a separate cold curve is calculated for each phase, and the transition pressure is determined from experiment.

Rice, McQueen, and Walsh showed that the cold curve can be determined from the shock Hugoniot, if a Mie-Grüneisen EOS is assumed.\(^{11}\) Using the Dugdale-MacDonald relation for the Grüneisen function,\(^{12}\) they calculated cold curves for 25 metals. Vaidya and Kennedy have made static compressibility measurements up to 45 kbar on 18 metals,\(^{13}\) and the results agree well with the shock wave predictions.\(^{14}\) To illustrate the procedure, let us write the Mie-Grüneisen EOS in the form\(^{15}\)

\[ P_C(\rho) = \rho \gamma(\rho) E_C(\rho) = P_H(\rho) - \rho \gamma(\rho) E_H(\rho). \tag{4} \]

\( P_H \) and \( E_H \) are the pressure and internal energy on the Hugoniot, which are obtained from experiment. The Grüneisen function, \( \gamma(\rho) \), is related to the Debye temperature, \( \Theta(\rho) \), by

\[ \gamma = \frac{d \ln \Theta}{d \ln \rho}. \tag{5} \]

The cold curve pressure, \( P_C \), and internal energy, \( E_C \), are related by

\[ P_C = \rho^2 \frac{d E_C}{d \rho}. \tag{6} \]
With these relationships, Eq. (4) can be integrated, as follows:

$$E_c(\rho) = \theta(\rho) \left[ E_c(\rho_o) \theta(\rho_o) + \int_{\rho_o}^{\rho} \frac{P_H(x) - x \gamma(x) E_H(x)}{x^2 \theta(x)} \, dx \right].$$  

(7)

If the functions $\gamma(\rho)$ and $\theta(\rho)$ are known explicitly, the cold curve can be determined quite easily from Eq. (7). If $\gamma(\rho)$ is expressed in terms of $P_c(\rho)$, as in the Dugdale-MacDonald formula, the procedure of Rice et al.\textsuperscript{11} may be used.

The above procedure is used in generating Sesame EOS tables as an alternative to Eq. (2). The technique is augmented by a high-density extrapolation, which defines the cold curve at densities that cannot be obtained by shock wave methods or in regimes where the Mie-Grüneisen EOS is not valid. An expression similar to Eq. (2), based upon the TFD EOS, is used in most cases.

For most applications the cold curve must be defined in tension ($\rho < \rho_o$) as well as in compression. The binding energy of the solid, $E_B$, is related to the pressure on the cold curve by an integral over the tension region,

$$E_B = -\int_{0}^{\rho_o} P_c(\rho) \rho^{-2} \, d\rho .$$  

(8)

For many materials, Eq. (2) gives reasonable values of the binding energy. This suggests that the Barnes form for the cold curve gives a realistic description for tension as well as compression. However, it is often desirable to attach another analytic form, such as one based upon a Lennard-Jones force law onto the cold curve at some density, and to adjust the constants so that the EOS agrees with experimental vaporization data. In this way, the experimental binding energy is usually matched exactly when generating a Sesame EOS.

B. Nuclear Contributions

For many solids the Debye model gives a reasonable estimate of the nuclear contributions to the EOS.
\[ P_N = \rho \gamma(\rho) R \left[ \frac{9}{8} \theta + 3D(\theta/T) \right], \]  

(9)

\[ D(x) = \frac{3}{x^3} \int_0^x \frac{y^3}{e^y - 1} \, dy, \]

where \( R \) is the gas constant. Equation (9) includes both the zero point and thermal contributions to the pressure. As mentioned above, \( \gamma(\rho) \) and \( \theta(\rho) \) are frequently estimated from the cold curve, using the Dugdale-MacDonald formula or a similar approximation. Experimentally, information about these parameters can be obtained from thermodynamic data, from reflected shock experiments, from measurements of the sound speed in a shocked material, and from shock measurements on foams.\(^{16}\) Sometimes simple analytic expressions for \( \gamma \) and \( \theta \) may be adequate to fit the data.

At high temperatures or low densities, the nuclear term should approach the ideal gas limit.

\[ P_N \to \rho RT \text{ as } T \to \infty \text{ or } \rho \to 0. \]  

(10)

Many schemes for estimating the nuclear contributions to the EOS are just interpolation formulas between Eqs. (9) and (10). One method used in Sesame is that of Kormer et al.\(^{17}\) A simplified version of this model has been used by Thompson and Lauson\(^{4}\) and by Merts and Magee.\(^{18}\) The reader should consult these references for discussion of the equations.

Barnes et al. have shown that a modified virial expansion of the nuclear term gives good results for the EOS of expanded materials.\(^{19,20}\) In their approach, Eq. (9) is used for compressed states, \( \rho > \rho_0 \). For \( \rho < \rho_0 \), they write

\[ P_N = RT \left[ \rho + B(T)\rho^2 + C(T)\rho^3 \right], \]  

(11)

where the functions \( B(T) \) and \( C(T) \) are determined by matching Eq. (11) smoothly to Eq. (9) at \( \rho = \rho_0 \). In more recent work, we have added another term to the expansion; the extra coefficient can be determined from the requirement that the energy approach that for an ideal gas as \( \rho \to 0 \).
When the nuclear term is added to the cold curve, it is found that the calculated isotherms have a Van der Waals form below some critical temperature. Hence these models lead to the formation of a two-phase region in which the vapor coexists with the condensed phase. In creating Sesame EOS tables, we attempt to vary the parameters in the model until the best fit to experimental vapor pressure measurements and other data is obtained.

In constructing a Sesame table, either the equilibrium or nonequilibrium results can be used in the mixed phase region. For problems in which a solid or liquid is heated rapidly and allowed to expand ("blowoff"), the correct treatment of vaporization can be essential in predicting hydrodynamic flow. Therefore, the equilibrium vapor pressures are normally used in constructing the high-temperature isotherms.

At low temperatures, however, vaporization may not occur on microsecond time scales. A solid can go into tension until critical stresses are exceeded, at which time spall (fracture) can take place. Therefore, some Sesame EOS tables are constructed with a negative pressure region at low temperatures, so that they can be used with spall models that are typical of hydrocodes.

The mixed phase region is particularly important in the analysis of fast reactor safety. In the reactor analysis code, SIMMER, both equilibrium and nonequilibrium thermodynamic variables are used in computing evaporation and condensation. We have developed a special Sesame EOS package for this code.

C. A Liquid Model - Melting

Although the empirical models discussed above can be made to match experimental vaporization data with good agreement, they do not predict a real melting transition. Melting is important, not only because it has an effect upon the EOS, but also because material strength disappears in the liquid phase. Consequently, the melting temperature at high pressures is needed for models of elastic-plastic flow.

We have developed a variation theory of liquids, which uses the hard sphere EOS with perturbation terms to correct for the effect of realistic interatomic forces. Our model is similar to that of Mansoori and Canfield. However, the interatomic interactions are obtained from the cold curve, instead of from pair potentials. The reader should consult Ref. 23 for details. This theory gives good results for hydrogen, deuterium, and argon. In particular, the predicted melting curve for argon was in good agreement with experiment. We are
working with this model to generate new Sesame EOS tables for metals, and the preliminary results are encouraging. Some calculations of the lead EOS are discussed in Sec. II.E.

D. Thermal Electronic Contributions

The term $P_e(\rho,T)$ in Eq. (1) represents contributions from thermal excitation and ionization of the electrons to the EOS. These effects become appreciable at temperatures above $10^4$ K, and they can be significant at much lower temperatures in special cases.

At the present time, most of the Sesame EOS tables are based upon the temperature-dependent TFD theory, using a code written by R. D. Cowan.\textsuperscript{9,25} Although the TFD theory neglects effects that result from discrete atomic levels and band structure, we believe it is adequate for reproducing some of the gross features of electronic excitation. Better theoretical methods are becoming available, and we intend to use them in generating Sesame tables in the future.

The calculations of Huebner et al. take the discrete atomic levels into account by solving the Schrödinger equation for the perturbed and thermally excited atoms.\textsuperscript{26} For temperatures less than $10^6$ K, experimental level data are used for those states that have been measured. At higher temperatures and for the highest densities, a mean ion model is used. Ion-ion correlations, which are neglected in these calculations, become important at high densities and low temperatures. For this region, we are investigating a new model, which has been proposed by Liberman.\textsuperscript{8}

E. An Example - Lead EOS

Shock Hugoniot data for lead\textsuperscript{27} provide some confirmation of the models. In Fig. 1, the experimental shock velocity ($U_s$) vs particle velocity ($U_p$) curve is compared with three calculations. The curve for the solid phase, based upon the Debye model, exhibits a linear $U_s$, $U_p$ relationship which is typical of metals. For strong shocks, there is a marked change of slope, which suggests that melting has occurred.\textsuperscript{28} Our calculated curve for the liquid agrees well with the experimental data. For still stronger shocks, there is sufficient thermal energy to cause significant electronic excitation and ionization. When this effect is included, even better agreement with experiment is obtained.

In summary, Sesame EOS tables are constructed from many different models and obtained from a variety of sources. When new theoretical and experimental results become available, the tables are revised or replaced, so that the best data is accessible to users.
III. STRUCTURE OF THE SESAME DATA LIBRARY

The Sesame library is a general purpose system for storing and accessing processed material property data. At the present time, Sesame only offers tables of thermodynamic properties, which is the purpose for which it was originally developed. However, it uses a flexible scheme for storing and cataloging data, so that it can be adapted to other applications. We plan to add opacity tables to the library soon.

For efficient storage and rapid data acquisition, the library is kept as a binary (unformatted) file on a disk or other mass storage device. Such a file is not portable, because binary files have different structures on different computing systems. To transmit the library to users, we have developed a procedure for writing the data onto a magnetic tape in a BCD card image format, which can be read and interpreted by most systems. This tape is preprocessed by the user at his own installation, using FORTRAN routines from the Sesame subroutine library. The result is a binary file, created by the local system, that is saved for future use. From time to time, the user can add or replace data on the library with update tapes from LASL.

A. Data Records

The fundamental unit of information on the Sesame library is a data record. One record contains a data string of arbitrary word length, written onto the file with an unformatted FORTRAN WRITE or similar output command. A description of the data records used in Sesame is given in Table I.

The type of data and specific format with which it is tabulated in a record is identified by a catalog number. For example, a 301 data record is used to tabulate pressure and internal energy as functions of density and temperature.
TABLE I

STRUCTURE OF THE SESAME DATA RECORDS

No. 201. Basic Data: five real words.

Z  - average atomic number
A  - average atomic weight
\(\rho_0\)  - solid density
\(B_0\)  - solid bulk modulus
\(C_{ex}\)  - exchange coefficient (not always used)

No. 301. EOS Tables: pressure and internal energy as functions of density and temperature.

NR, NT, (R(I), I = 1, NR), (T(J), J = 1, NT), ((P(I,J), I = 1, NR), J = 1, NT)
NR  - number of points on density mesh (real number)
NT  - number of points on temperature mesh (real number)
R   - density mesh
T   - temperature mesh
P   - pressure
E   - internal energy

No. 302. Inverted EOS tables: pressure and temperature as functions of density and internal energy.

NR, NE, (R(I), I = 1, NR), (ET(J), J = 1, NE), (EC(I), I = 1, NR),
((P(I,J), I = 1, NR), J = 1, NE), ((T(I,J), I = 1, NR), J = 1, NE)
NR  - number of points on density mesh (real number)
NE  - number of points on thermal energy mesh (real number)
R   - density mesh
ET  - thermal energy mesh (ET = E - EC)
EC  - internal energy at 0 K
P   - pressure
T   - temperature
TABLE I (cont)

No. 401. Vaporization Tables: thermodynamic quantities on the vapor-liquid and vapor-solid coexistence curve.

- \( NT, (P(I), I = 1, NT), (T(I), I = 1, NT), (RG(I), I = 1, NT), \)
  \( (RL(I), I = 1, NT), (EG(I), I = 1, NT), (EL(I), I = 1, NT) \)

- \( NT \)  - number of temperatures (real number)
- \( P \)  - vapor pressure
- \( T \)  - temperature
- \( RG \)  - vapor density on coexistence line
- \( RL \)  - density of liquid or solid on coexistence line
- \( EG \)  - internal energy on coexistence line
- \( EL \)  - energy of liquid or solid on coexistence line

- When interpolating on a 401 data string, all quantities except \( EG \) can be used as independent variables.
- \( T(NT) \) is the critical temperature.

---

*Units used in the Sesame library are*

- pressure - GPa
- energy - MJ/kg
- density - Mg/m³
- temperature - K.

---

\[
P = f_1(\rho, T),
\]

\[
E = f_2(\rho, T).
\]

A 302 data record is used to tabulate the "inverted" EOS, pressure and temperature as functions of density and energy.
\[ P = g_1(\rho, E), \quad T = g_2(\rho, E). \] (13)

Equations (13) could be calculated from a 301 data record, but note that the formats are different in the two cases. At the present time, the 302 data records are not kept on the library, because they are easy to generate from the 301 form when they are needed.

When a new data form is added to the library, a format is worked out and it is assigned a catalog number. Hence the library will become more general in time. There are only two restrictions on the formats that can be used. Each record must be all Hollerith, all real, or all integer, and all Hollerith records must be assigned a catalog number from 100-199.

B. Material File

A collection of several data records, together with an index record, forms a material file, as illustrated in Fig. 2. Each file contains data for a particular material, which is identified by a material number from 1000-9999. A list of the Sesame materials presently available in the library is given in Table II. Those materials identified as "reactor safety" were generated for the SIMMER-II code, and they cover a more limited density-temperature range than the others. Materials 5760, 7150, and 7151 were obtained from Lawrence Livermore Laboratory (LLL). The other tables were generated at LASL. Some of these tables are being revised, and new materials will be added continually to the library.

The index record to a file gives the material number, the number of data records in the file, and the catalog number and word length of each record. The structure of an index record is described in Table III.

C. Structure of the Library

The library is a collection of material files, preceded by a directory, as illustrated in Fig. 2. Files are separated by END FILE (EOF) marks, which can be used to search through the library for a particular material. On IBM, and on other systems for which multiple file-sets are cumbersome, the EOF marks can be replaced by one-word ASCII records. Directions for making this simple change are included in the instructions that are enclosed with the Sesame subroutine library.

The directory file, described in Table IV, gives the number of material files in the library and the material I.D. number of each.
IV. THE INTERPOLATION ALGORITHM AND SUBROUTINE LIBRARY

The Sesame subroutine library was developed to simplify use of the tables. It contains routines to preprocess the magnetic BCD data tape, to load data from the library into local storage, and to search and interpolate on the data. With the exception of binary input/output, which has been localized in only two routines, all subprograms are written in standard FORTRAN, so that no system conversion should be required. Use of the subroutines is discussed in comment cards and in the instructions that are provided with the subroutine library. Hence we will discuss only general concepts in this report.

A. Rational Function Interpolation

Considerable work has been done to develop an interpolation algorithm suitable for EOS data. In some applications where accuracy can be sacrificed, linear interpolation is attractive because it is simple and computationally fast. Consequently, we provide users with a linear scheme as one of the options.

However, we found that it was necessary to develop a higher order method as well. The Sesame tables often cover wide temperature and density ranges, and the mesh is often much too coarse for a linear algorithm. Moreover, many users require derivatives of the functions (sound speed, specific heat), and linear methods give poor estimates of these quantities.

Our rational function method of interpolation is particularly useful for functions having a rapid or discontinuous change in the derivatives. This feature is important in working with EOS data, which may have phase transitions.
TABLE II
UNCLASSIFIED SESAME MATERIALS AS OF DECEMBER 1977

<table>
<thead>
<tr>
<th>Code</th>
<th>Substance</th>
<th>Code</th>
<th>Substance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1540</td>
<td>Uranium(^a)</td>
<td>5760</td>
<td>Helium(^b)</td>
</tr>
<tr>
<td>2020</td>
<td>Beryllium(^a)</td>
<td>7081</td>
<td>Boron Carbide (reactor safety)</td>
</tr>
<tr>
<td>2140</td>
<td>Iron(^a)</td>
<td>7111</td>
<td>Nevada Alluvium(^a)</td>
</tr>
<tr>
<td>2145</td>
<td>Iron (reactor safety)</td>
<td>7150</td>
<td>Water(^b)</td>
</tr>
<tr>
<td>2448</td>
<td>Sodium (reactor safety)</td>
<td>7151</td>
<td>Steam(^b)</td>
</tr>
<tr>
<td>2700</td>
<td>Gold(^a)</td>
<td>7170</td>
<td>CH(_2) (polyethylene)(^a)</td>
</tr>
<tr>
<td>2980</td>
<td>Molybdenum(^a)</td>
<td>7240</td>
<td>(\text{^6LiD})!</td>
</tr>
<tr>
<td>3100</td>
<td>Nickel(^a)</td>
<td>7370</td>
<td>(\text{^6LiH})!</td>
</tr>
<tr>
<td>3200</td>
<td>Lead(^a)</td>
<td>7380</td>
<td>(\text{SiO}_2) (quartz, sand)(^a)</td>
</tr>
<tr>
<td>3330</td>
<td>Copper(^a)</td>
<td>7390</td>
<td>Westerly granite(^a)</td>
</tr>
<tr>
<td>3710</td>
<td>Aluminum(^a)</td>
<td>7410</td>
<td>(\text{Al}_2\text{O}_3) (alumina, sapphire)(^a)</td>
</tr>
<tr>
<td>3730</td>
<td>Platinum(^a)</td>
<td>7432</td>
<td>(\text{UO}_2) (reactor safety)</td>
</tr>
<tr>
<td>4100</td>
<td>Brass(^a)</td>
<td>7520</td>
<td>Vermiculite (mica)(^a)</td>
</tr>
<tr>
<td>4270</td>
<td>Stainless Steel(^a)</td>
<td>7560</td>
<td>Urethane(^a)</td>
</tr>
<tr>
<td>5263</td>
<td>Deuterium</td>
<td>7590</td>
<td>CH (polystyrene)(^a)</td>
</tr>
<tr>
<td>5410</td>
<td>Neon(^a)</td>
<td>8180</td>
<td>High Explosive (XO-290)(^c)</td>
</tr>
</tbody>
</table>

\(^a\)Generated by J. F. Barnes.
\(^b\)Obtained from LLL.
\(^c\)Generated by C. L. Mader.

Finally, we have studied methods of optimizing the density and temperature mesh, and we feel that current storage requirements can be cut in half, eventually.

B. Accessing Data from the Data Library

The subroutine file contains a number of routines that search the data library for a material requested by the user, load data into a local array, and
TABLE III
STRUCTURE OF THE INDEX RECORD FOR A SESAME MATERIAL FILE

MATID, DATE1, DATE2, VERS, NREC, (TBLID(I), I = 1, NREC), (NWDS(I), I = 1, NREC)
MATID - material I.D. number (real number)
DATE1 - date of creation of material
DATE2 - date when material file was last updated
VERE - version number of most recent update
NREC - number of data records (real number)
TBLID - table of catalog numbers for the data records, in the same order as they
are stored in the material file
NWDS - for each TBLID(I), NWDS(I) is the number of words in the data record
(real number)

TABLE IV
STRUCTURE OF THE DIRECTORY FILE FOR THE SESAME LIBRARY

Record 1. Consists of three words.
   N, DATE, VERS

N  - number of material files in the library (real number)
DATE - date of current version
VERE - version number

Record 2. Consists of two arrays.
   (MATID(I), I = 1, N), (NWDS(I), I = 1, N)
MATID - table of material I.D. numbers, in the same order as the material files
in the library (real numbers)
NWDS - for each material file MATID(I), NWDS(I) is the number of words in the
index record
compute thermodynamic functions, Eqs. (12) and (13), by search and interpolation. These programs are particularly useful for hydrodynamic codes. They provide for computations involving several Sesame tables, for the use of Sesame tables along with other EOS options, and for the specification of the same Sesame table in more than one region. All the necessary bookkeeping is internal to the routines. These programs have been checked out and installed into the one-dimensional hydrocode SIN.21 Sesame tables can be used with both LaGrangian and Eulerian codes, in problems that involve elastic-plastic flow, spall, foams, nonequilibrium phase transitions, and radiation flow.

For some reactor safety applications, EOS data is used in modeling multiphase fluid flow and phase transitions. The SIMMER code solves coupled sets of hydrodynamic equations for the solid, liquid, and a vapor mixture of varying chemical composition.22 A special Sesame EOS package has been written for this problem. An algorithm was developed to calculate the EOS of the vapor mixture, as a function of its chemical composition, from EOS tables for the pure components. The tabular package also provides liquid-vapor coexistence quantities that are used in computing evaporation and condensation rates. These specialized routines are not available from the subroutine library, but interested users can send for them.

C. A Display Package

We have developed a new code to display Sesame EOS data graphically, in a variety of formats. The existing version of the code has several system-dependent features (primarily input/output) that limit its use to LASL computers. However, it is a modular code, designed to be adapted to other computer systems and display devices with minimum effort.

The heart of the display code is a plotting module that can produce graphs on three different output devices in use at LASL. Most of the plotting subroutines are independent of the hardware to be used in graphing. Those routines that refer to a specific output device are limited to elementary graphing operations (plotting a point or drawing a vector). Consequently, the code can be made to plot on new output devices without major revision.

The data to be graphed by the plotting module is generated by another module, which uses the interpolation routines discussed above. With relatively few commands, a user can generate isotherms, isentropes, and Hugoniot curves. Data can be plotted on either linear or logarithmic axes, and the user can zoom in on a
specific region of interest. Many curves can be plotted on the same figure, and each curve is labeled for identification.

Although the code is not yet available outside LASL, the subroutines that are used to calculate Hugoniotls and isentropes are included in the Sesame subroutine library, for interested users.

V. SUMMARY AND CONCLUSIONS

In this report, we have given an overview of the Sesame library. Both the theory and the data handling procedures were described. We have not presented an evaluation of specific EOS tables and compared with experiment. Such studies will be reported elsewhere. Detailed discussion of computer programs has been avoided. However, users will be able to obtain much information from comment cards given in the routines themselves and from the instruction sheets that are enclosed with the listings and magnetic tapes.

As we continue to add new data and make improvements to the Sesame library, users will be kept informed by reports and newsletters.

ACKNOWLEDGMENTS

The authors are grateful to John F. Barnes, Jr., whose work formed the foundation of Sesame, and who generated many of the EOS tables in the library. We wish to thank Charles A. Forest, who contributed much useful advice in setting up the data storage and retrieval procedures. We are grateful to our colleagues at LLL, who have provided us with their data. There are many other individuals, too numerous to mention, who have aided us along the way.

REFERENCES


8. D. A. Liberman, to be published.


HYDSES: A Subroutine Package for Using
Sesame in Hydrodynamic Codes

J. Abdallah, Jr.
G. I. Kerley
B. I. Bennett
J. D. Johnson
R. C. Albers
W. F. Huebner
HYDSES: A SUBROUTINE PACKAGE
FOR USING SESAME IN HYDRODYNAMIC CODES

by

J. Abdallah, Jr., G. I. Kerley, B. I. Bennett,
J. D. Johnson, R. C. Albers, and W. F. Huebner

ABSTRACT

HYDSES is a subroutine package developed for using Sesame equation-
of-state and opacity data in hydrodynamic codes. This package is faster,
more space efficient, and includes more capability than the previous
package. The HYDSES package has been used successfully in
hydrodynamic calculations. This report describes the package and its use in
hydrodynamic codes.

I. INTRODUCTION

The Sesame library is a file of equation-of-state (EOS) and other material properties that can be ac-
cessed and used by application computer programs in need of such data. There also is a set of library-
associated subroutines\textsuperscript{1,2} that can be used in the ap-
plication programs to facilitate the retrieval of
Sesame data. HYDSES is a new subroutine package
including many new features and improvements.
This report discusses HYDSES and its use in appli-
cation programs.

For most cases the use of Sesame data within a
code is a two-phase process. Phase one is initialization
or setup. This phase consists of retrieving the
desired data table from the disk, converting it into the
proper format, storing it into a block of core, and
saving its location for subsequent use. After this is
done, it may be desirable to perform a one-time
initial-state computation based on various sets of in-
dependent variables. Initialization is generally a
one-pass phase that requires a small amount of time
compared to the total run time of the application.
After initialization the application program is ready
for phase two, the computational phase. Here the
application program cycles and enters the Sesame
routines that perform interpolation\textsuperscript{3} on the Sesame
data tables an enormous number of times. The com-
putational phase is usually the time-consuming
phase.

A. HYDSES Features

The following features have been incorporated
into the HYDSES package.

(1) HYDSES is fast. New computational-phase
Sesame subroutines (T4DAT, T4DATI, T4INTP,
RATFN1) have been developed to provide increased
performance. The new features include the follow-
ing.

a. Recomputation of interpolation coefficients is
   avoided if possible.

b. Computational-phase Sesame subroutines all
   avoid the use of calling sequences. Instead,
parameters are now passed through common
blocks. This method of programming was
adopted to improve computational speed.
because of a large percentage of time was spent in address setup for calling parameters.

c. Coding internal to the interpolation package (T4INTP, T4DAT, T4DATI) was optimized.

In addition to the time-saving features introduced into the computational-phase subroutines, random I/O is used in the HYDSES package to speed up the initialization-phase subroutines (see Appendix A). The use of random I/O eliminates the need for time-consuming sequential searches through the Sesame library file. An optional sequential I/O capability has been provided (see Appendix A) for users who encounter difficulties adapting this random I/O method to their computer system.

(2) HYDSES is space efficient. Dependent variables of Sesame tables are double-packed into the data block by initialization routines after being read into core. This reduces the space needed to store a table by a factor just less than two. In addition, HYDSES provides routines (WINDOW, WDWCHK, ISRCH) to window out and verify unwanted portions of a given temperature-based table. These routines can be used to cut down on storage space when only a certain density and temperature range is required by the application program.

(3) HYDSES can accommodate multiple data types. Because types of data other than EOS are becoming available, the package includes new initialization-phase subroutines to access radiative Rosseland mean opacities\(^4\) (GETRPO, GETRTL), Planck mean opacities\(^5\) (GETRPO), total (radiative and thermal conduction) opacities\(^5\) (GETRTL), thermal conductivity of electrons\(^6\) (GETTCE), ion charge (number of free electrons)\(^7\) (GETNFE), and two-temperature equation-of-state tables.\(^8\) These subroutines can be adapted easily into various application programs. They are also written in a modular manner so that if a new data table becomes available, the associated initialization subroutine would be easy to code.

(4) Several utility subroutines (T4EOSA, T4RTPE, T4PTRE) have been introduced to enhance capabilities when inverted (energy-based) EOS tables are used. Inverted tables are stored so that the dependent variables are \(P\) and \(T\) and the independent variables are \(\rho\) and \(E\), whereas the non-inverted tables are temperature-based and corre-
pond to \(P(\rho, T)\) and \(E(\rho, T)\). Here \(\rho\), \(T\), \(P\), and \(E\) are the thermodynamic quantities of density, temperature, pressure, and energy. The temperature-based tables are obtained by using GETEOS, and the inverted tables are obtained by using GETINV. Temperature-based tables exist in the Sesame library. They are converted to the inverted form by INV301.

Two of the new subroutines (T4RTPE and T4PTRE) are intended primarily for (but not restricted to) initial-state determination when inverted tables are used. T4RTPE computes \(P(\rho, T)\) and \(E(\rho, T)\), whereas T4PTRE computes \(\rho(P, T)\) and \(E(P, T)\). A third new subroutine (T4EOSA) may be used in the computational phase to scale the EOS and/or to treat foamed materials and special phase transitions.

B. System-Dependent Features

The package presented here is designed for use on the Los Alamos Scientific Laboratory (LASL) CDC 7600 LTSS computers using the FTN/LTSS compiler and large-core memory (LCM). The system-dependent features of HYDSES include double-packing and random I/O. All random I/O is isolated to subroutine INBUF, which calls the system-dependent RDISK subroutine. All double-packing and unpacking is performed through the system-dependent in-line shift function that assumes 60-bit words. Use of the shift function is scattered throughout the package, whereas all double-packing is performed through the HYDSES function DPACK. The user should be aware of the Level 2 statements used for LCM declaration.

C. Summary

Table I summarizes all the subroutines appearing in the HYDSES package. Section II describes the initialization-phase subroutines, Sec. III describes the computational-phase subroutines, Sec. IV describes the general use of the package, Sec. V presents a test example, and Sec. VI describes how the HYDSES package may be obtained.
<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Type</th>
<th>Phase</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>T4DAT</td>
<td>U</td>
<td>LC</td>
<td>Interpolate on T-based tables</td>
</tr>
<tr>
<td>.GETRPO</td>
<td>U</td>
<td>I</td>
<td>Get radiative Rosseland/Planck table</td>
</tr>
<tr>
<td>GETTCE</td>
<td>U</td>
<td>I</td>
<td>Get thermal conductivity of electrons</td>
</tr>
<tr>
<td>GETNFE</td>
<td>U</td>
<td>I</td>
<td>Get mean ion charge table</td>
</tr>
<tr>
<td>MATCHK</td>
<td>A</td>
<td>I</td>
<td>Checks to see if a material has been loaded</td>
</tr>
<tr>
<td>TABFCH</td>
<td>A</td>
<td>I</td>
<td>Fetch table from library (random I/O)</td>
</tr>
<tr>
<td>INBUFR</td>
<td>A</td>
<td>I</td>
<td>Random I/O Read</td>
</tr>
<tr>
<td>DPACK</td>
<td>A</td>
<td>I</td>
<td>Double-packs two real numbers</td>
</tr>
<tr>
<td>GETEOS</td>
<td>U</td>
<td>I</td>
<td>Get EOS table</td>
</tr>
<tr>
<td>GETNUC</td>
<td>U</td>
<td>I</td>
<td>Get nuclear EOS table</td>
</tr>
<tr>
<td>SUBNUC</td>
<td>U</td>
<td>I</td>
<td>Subtract nuclear EOS from EOS</td>
</tr>
<tr>
<td>IONEOS</td>
<td>A</td>
<td>I</td>
<td>EOS from Cowan model</td>
</tr>
<tr>
<td>WINDOW</td>
<td>U</td>
<td>I</td>
<td>Window T-based tables</td>
</tr>
<tr>
<td>ISRCH</td>
<td>A</td>
<td>I</td>
<td>Index search</td>
</tr>
<tr>
<td>WDWCHK</td>
<td>U</td>
<td>I</td>
<td>Window check routine</td>
</tr>
<tr>
<td>GETRTL</td>
<td>U</td>
<td>I</td>
<td>Get Rosseland/Total table</td>
</tr>
<tr>
<td>ISRCHK</td>
<td>U</td>
<td>I</td>
<td>Index search</td>
</tr>
<tr>
<td>T4PTRE</td>
<td>U</td>
<td>I</td>
<td>Computes s,E from P,T from inverted table</td>
</tr>
<tr>
<td>T4INTP</td>
<td>A</td>
<td>I,C</td>
<td>Two-dimensional interpolator</td>
</tr>
<tr>
<td>GETINV</td>
<td>U</td>
<td>I</td>
<td>Get inverted EOS table</td>
</tr>
<tr>
<td>RATFNI</td>
<td>A</td>
<td>I,C</td>
<td>One-dimensional interpolator</td>
</tr>
<tr>
<td>T4DATI</td>
<td>U,A</td>
<td>I,C</td>
<td>Interpolate on inverted table</td>
</tr>
<tr>
<td>T4RTPE</td>
<td>U</td>
<td>I</td>
<td>Compute, P(s,T),E(s,T) from inverted table</td>
</tr>
<tr>
<td>INV301</td>
<td>A</td>
<td>I</td>
<td>Invert 301 table</td>
</tr>
<tr>
<td>T4EOSA</td>
<td>U</td>
<td>I,C</td>
<td>Computes a scaled EOS and includes a phase transition</td>
</tr>
<tr>
<td>TABSEQ</td>
<td>A</td>
<td>I</td>
<td>Fetch table from library (sequential I/O)</td>
</tr>
<tr>
<td>INBUF</td>
<td>A</td>
<td>I</td>
<td>Sequential read</td>
</tr>
</tbody>
</table>

*U* means that the routine is intended to be called directly by the user program. An *A* means that the routine is an auxiliary routine called by HYDSES subroutines.

*C* stands for computational phase, and *I* stands for initialization phase.
II. INITIALIZATION-PHASE SUBROUTINES

All HYDSES subroutines are used in the initialization phase. However, several of them are intended primarily for use in the computational phase. Detailed descriptions of these subroutines are presented in Sec. III, and their use in the initialization phase is summarized in Sec. II.D.

Section II.A describes the initialization-phase "get" subroutines that are important to all HYDSES users.

Section II.B describes several optional utility subroutines that are useful for "windowing" data tables and for obtaining electronic EOS tables.

Section II.C describes subroutines that provide initial-state computation from inverted tables.

Section II.D describes utility subroutines that are called by those discussed in II.A, II.B, and II.C. Most users will have no need for the material presented in II.D.

A. HYDSES "get" Subroutines

The primary purpose of the HYDSES "get" subroutines is to load data from Sesame library disk files into data arrays. Each "get" subroutine corresponds to different types of data. Only those "get" subroutines that correspond to data required by the user application program need be considered. The internal operation of a typical "get" subroutine is outlined here.

1. The "get" subroutine checks to see if the requested data were loaded previously. This check is done to avoid loading identical data tables.
2. The appropriate data are read in from the Sesame library disk file.
3. The data are transformed to the appropriate data type and/or units. For example, the electron thermal conductivity is computed from the electron conduction opacity. The electron conduction opacity is stored in the Sesame library.
4. Dependent variables are double-packed.
5. The location of the loaded data is saved for later use.

Table II summarizes the various "get" subroutines and their associated calling sequences. Here IDT is a data-type number 1, 2, 3, ... etc., assigned by the application program to the various table types. Thus, for one application IDT may be set to 1 for EOS tables, to 2 for Rosseland/Planck opacity tables, and to 3 for inverted EOS tables. The IDT flag is used later in the computational phase to locate the desired data upon interpolation. This feature allows the use of multiple Sesame table types in a given application program.

Table III summarizes the tables obtained by the HYDSES "get" routines. Simple modifications may be made to these routines to obtain the tables in other desired units. Note that subroutines GETEOS, GETNUC, and GETRPO are coded to give units for the dependent variables that are independent of mass. To obtain the mass-dependent units, the factor of density in the unit conversion of these routines should be eliminated. Note that subroutines GETEOS and GETINV return the normal density ($\rho_n$) in TBLS (LCNT+1). Table IV shows the detailed format of Sesame data tables in array TBLS upon return from the "get" subroutines. Table V presents the format of the Sesame data tables used by HYDSES as they appear in the Sesame library files. See Appendix A for the Sesame library file structure.

B. Initialization-Phase Utility Subroutines

1. WINDOW. This subroutine is used to reduce the size of a temperature-based data table when the full Sesame density and temperature range is not required by the applications program. If windowing is desired, the corresponding "get" subroutine can be modified easily to call WINDOW. The WINDOW calling sequence is discussed next.

CALL WINDOW (DSTR,RLIM,TZRO,TLIM, LOC,NTAB)

DSTR(INPUT/OUTPUT) - LCM data string starting at NR (see Table V).

RLIM(INPUT) - Array of lower and upper bounds on density.

TZRO(INPUT) - Temperature floor. A low temperature near TZRO will appear in the windowed tables if possible.

TLIM(INPUT) - Array of lower and upper bounds on temperature.

LOC(OUTPUT) - New length of data table. It should be used to adjust LCNT for the next table in the associated "get" routine.
<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Calling Sequence</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>GETEOS</td>
<td>IR,MID,IDT,TBLS,LCNT,LU,IFL,ZB</td>
<td>Get EOS tables</td>
</tr>
<tr>
<td>GETINV</td>
<td>IR,MID,IDT,TBLS,LCNT,LU,IFL,ZB</td>
<td>Get inverted EOS tables</td>
</tr>
<tr>
<td>GETNUC</td>
<td>IR,MID,IDT,TBLS,LCNT,LU,IFL</td>
<td>Get nuclear (ion core) EOS tables</td>
</tr>
<tr>
<td>GETRPO</td>
<td>IR,MID,IDT,TBLS,LCNT,LU,IFL</td>
<td>Get radiative Rosseland Planck opacity tables</td>
</tr>
<tr>
<td>GETTCE</td>
<td>IR,MID,IDT,TBLS,LCNT,LU,IFL</td>
<td>Get electronic thermal conductivity tables</td>
</tr>
<tr>
<td>GETNFE</td>
<td>IR,MID,IDT,TBLS,LCNT,LU,IFL</td>
<td>Get ion charge (number of free electron) tables</td>
</tr>
<tr>
<td>GETRRL</td>
<td>IR,MID,IDT,ITOT,TBLS,LCNT,LU,IFL</td>
<td>Get the radiative Rosseland and/or total opacity tables</td>
</tr>
</tbody>
</table>

**Calling Parameter Definitions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IR</td>
<td>INPUT</td>
<td>Material region number for which data is to be loaded</td>
</tr>
<tr>
<td>MID</td>
<td>INPUT</td>
<td>Sesame material identification number (id) for data to be loaded</td>
</tr>
<tr>
<td>IDT</td>
<td>INPUT</td>
<td>Data type to be assigned to loaded data</td>
</tr>
<tr>
<td>TBLS</td>
<td>OUTPUT</td>
<td>LCM array where tables are to be stored</td>
</tr>
<tr>
<td>LCNT</td>
<td>INPUT/OUTPUT</td>
<td>Position in TBLS where data are to be stored</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Position for storing next table</td>
</tr>
<tr>
<td>LU</td>
<td>INPUT</td>
<td>Sesame library unit No.</td>
</tr>
<tr>
<td>IFL</td>
<td>OUTPUT</td>
<td>Flag. IFL &lt; 0 means that IFL extra words are needed in TBLS; IFL = 0 means the requested data could not be located; IFL = 1 means that the requested data were loaded successfully; IFL = 2 means that the data were previously loaded</td>
</tr>
<tr>
<td>ZB(1)</td>
<td>OUTPUT</td>
<td>Average atomic number (number weighted mean z)</td>
</tr>
<tr>
<td>ZB(2)</td>
<td>OUTPUT</td>
<td>Z^2</td>
</tr>
<tr>
<td>ZB(3)</td>
<td>OUTPUT</td>
<td>Atomic mass (number weighted mean)</td>
</tr>
<tr>
<td>ITOT</td>
<td>INPUT</td>
<td>ITOT = 0 for radiative Rosseland table only; ITOT = 1 for both radiative Rosseland and total Rosseland opacity tables</td>
</tr>
</tbody>
</table>
**TABLE III**

**HYDSES 'GET' ROUTINE SUMMARY**

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Tables</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>GETEOS</td>
<td>P/E(ρ, T)</td>
<td>P in 10^4 J/cm^2, E in 10^4 J/cm^2, ρ in g/cm^4, T in keV</td>
</tr>
<tr>
<td>GETINV</td>
<td>P/T(ρ, E)</td>
<td>P in Mbar, E in Mbar·cm^3/g, ρ in g/cm^4, T in K</td>
</tr>
<tr>
<td>GETNUC</td>
<td>P_N/E_N(ρ, T)</td>
<td>P_N in 10^4 J/cm^2, E_N in 10^4 J/cm^2, ρ in g/cm^4, T in keV</td>
</tr>
<tr>
<td>GETRPO</td>
<td>κ_R/κ_p(ρ, T)^a</td>
<td>κ_R in cm^-1, κ_p in cm^-1, ρ in g/cm^4, T in keV</td>
</tr>
<tr>
<td>GETTCE</td>
<td>v_e/0(ρ, T)^a</td>
<td>v_e in 10^4 J/(keV·cm·10^-8 s), ρ in g/cm^4, T in keV</td>
</tr>
<tr>
<td>GETNFE</td>
<td>N_f/0(ρ, T)^a</td>
<td>N_f dimensionless, ρ in g/cm^4, T in keV</td>
</tr>
<tr>
<td>GETRTL</td>
<td>κ_R/κ_T(ρ, T)^a</td>
<td>κ_R in cm^2/g, κ_T in cm^2/g, ρ in g/cm^4, T in keV</td>
</tr>
</tbody>
</table>

**Legend:**

P is total pressure,
E is total internal energy,
ρ is density,
T is temperature,
P_N is nuclear pressure,
E_N is nuclear internal energy,
κ_R is radiative Rosseland mean opacity,
κ_p is Planck opacity,
v_e is thermal conductivity of electrons,
N_f is the number of free electrons/atom, and
κ_T is total Rosseland opacity.

---

^aAll quantities are actually stored as base 10 logarithms U/g.

x/y indicates double-packed dependent variables (y = 0 means zero is stored in the second half-word).

(x,y) indicates independent variables.

1 jerk = 10^8 J.

1 shake = 10^-5 s.
TABLE IV
FORMAT OF SESAME TABLES IN ARRAY TBLS

Temperature-Based Tables

<table>
<thead>
<tr>
<th>Index</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCNT</td>
<td>Material id</td>
</tr>
<tr>
<td>LCNT+1</td>
<td>( \rho_0 ) for EOS or inverted EOS; IDT otherwise</td>
</tr>
<tr>
<td>LCNT+2</td>
<td>Number of densities (NR)</td>
</tr>
<tr>
<td>LCNT+3</td>
<td>Number of temperatures (NT)</td>
</tr>
<tr>
<td>LCNT+4</td>
<td>NR densities</td>
</tr>
<tr>
<td>LCNT+4+NR</td>
<td>NT temperatures</td>
</tr>
<tr>
<td>LCNT+4+NR+NT</td>
<td>NR*NT double-packed dependent variables (see Table III)</td>
</tr>
</tbody>
</table>

Energy-Based EOS Tables

<table>
<thead>
<tr>
<th>Index</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCNT</td>
<td>Material id</td>
</tr>
<tr>
<td>LCNT+1</td>
<td>( \rho_0 )</td>
</tr>
<tr>
<td>LCNT+2</td>
<td>Number of densities (NR)</td>
</tr>
<tr>
<td>LCNT+3</td>
<td>Number of energies (NE)</td>
</tr>
<tr>
<td>LCNT+4</td>
<td>NR densities</td>
</tr>
<tr>
<td>LCNT+4+NR</td>
<td>NE energy differences ([E(\rho_0,T_i) - E(\rho_0,T_i), i = 1, NE])</td>
</tr>
<tr>
<td>LCNT+4+NR+NE</td>
<td>NR low-temperature energies ([E(\rho_0,T_i), i = 1, NR])</td>
</tr>
<tr>
<td>LCNT+4+NR+NE+NR</td>
<td>NE*NR double-packed P/T values</td>
</tr>
</tbody>
</table>

*All values are stored as floating-point numbers.
### Table V

**FORMAT OF SESAME TABLES IN LIBRARY**

**Table 201 - Basic Data**

<table>
<thead>
<tr>
<th>Word #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Average atomic number (number weighted mean $z$)</td>
</tr>
<tr>
<td>2</td>
<td>Mean atomic mass ($a$)</td>
</tr>
<tr>
<td>3</td>
<td>Normal (solid) density $\rho_a$</td>
</tr>
<tr>
<td>4</td>
<td>Solid bulk modulus ($= 0$ for material id's &gt; 9999)</td>
</tr>
<tr>
<td>5</td>
<td>Exchange coefficient ($= 0$ for material id's &gt; 9999)</td>
</tr>
</tbody>
</table>

**Table 301 - EOS Tables**

<table>
<thead>
<tr>
<th>Word #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Number of densities (NR)</td>
</tr>
<tr>
<td>2</td>
<td>Number of temperatures (NT)</td>
</tr>
<tr>
<td>3</td>
<td>NR densities (Mg/m$^3$)</td>
</tr>
<tr>
<td>NR+3</td>
<td>NT temperatures (K)</td>
</tr>
<tr>
<td>NR+NT+3</td>
<td>NR*NT pressures (GPa): ordered with density index increasing fastest</td>
</tr>
<tr>
<td>NR+NT+3</td>
<td>NR*NT energies (MJ/kg): ordered with density index increasing fastest</td>
</tr>
</tbody>
</table>

**Table 502 - Radiative Rosseland/Planck Opacity Table**

<table>
<thead>
<tr>
<th>Word #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Number of densities (NR)</td>
</tr>
<tr>
<td>2</td>
<td>Number of temperatures (NT)</td>
</tr>
<tr>
<td>3</td>
<td>NR (log$_{10}$) densities (g/cm$^3$)</td>
</tr>
<tr>
<td>3+NR</td>
<td>NT (log$_{10}$) temperatures (eV)</td>
</tr>
<tr>
<td>3+NR+NT</td>
<td>NR*NT double-packed (log$_{10}$) Rosseland and Planck opacities (cm$^2$/g) ordered with density index increasing fastest</td>
</tr>
</tbody>
</table>

**Table 503 - Electron Conduction Opacity Table**

<table>
<thead>
<tr>
<th>Word #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Number of densities (NR)</td>
</tr>
<tr>
<td>2</td>
<td>Number of temperatures (NT)</td>
</tr>
<tr>
<td>3</td>
<td>NR (log$_{10}$) densities (g/cm$^3$)</td>
</tr>
<tr>
<td>3+NR</td>
<td>NT (log$_{10}$) temperatures (eV)</td>
</tr>
<tr>
<td>3+NR+NT</td>
<td>NR*NT (log$_{10}$) electron conductive opacities (cm$^2$/g) ordered with density index increasing fastest</td>
</tr>
</tbody>
</table>

**Table 504 - Ion Charge (Number Of Free Electrons) Table**

<table>
<thead>
<tr>
<th>Word #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Number of densities (NR)</td>
</tr>
<tr>
<td>2</td>
<td>Number of temperature (NT)</td>
</tr>
<tr>
<td>3</td>
<td>NR (log$_{10}$) densities (g/cm$^3$)</td>
</tr>
<tr>
<td>3+NR</td>
<td>NT (log$_{10}$) temperatures (eV)</td>
</tr>
<tr>
<td>3+NR+NT</td>
<td>NR*NT (log$_{10}$) mean ion charges (number of free electrons/atom) ordered with density index increasing fastest</td>
</tr>
</tbody>
</table>

Notes: Table 301 appears for material id's ≤ 9999 only.
Tables 502, 503, and 504 appear for material id's ≥ 9999 only.
In general, the opacity material id = 10000 + Sesame EOS material id.
NTAB(INPUT) - Number of data tables appearing in the data string. (A double-packed table counts as one).

WINDOW can be used to reduce storage of inverted tables if applied before inversion occurs in GETINV. Note that all densities and temperatures should be in the units appearing in the data string, and they should be logarithmic if the table is logarithmic.

2. WDWCHK. This subroutine is used to see if a temperature-based table falls within user-specified limits.

CALL WDWCHK(NRN,IDT,TBLS,RWDW, TWDW,RTBL,TTBL,IWWDW)

NRN(INPUT) - Material region number.
IDT(INPUT) - Data type of temperature-based table.
TBLS(INPUT) - LCM table-storage area.
RWDW(INPUT) - Array of lower and upper density limits.
TWDW(INPUT) - Array of lower and upper temperature limits.
RTBL(OUTPUT) - Lower and upper density limits of table.
TTBL(OUTPUT) - Lower and upper temperature limits of table.
IWWDW(OUTPUT) = 0 if RWDW or TWDW fall within table limits and = 1 if RWDW or TWDW are outside table limits.

This routine may be used any time after the table is loaded.

3. SUBNUC. This subroutine is used when two-temperature equations of state are used.* If, for example, the user wants separated electronic and nuclear (ion core) EOS tables, GETEOS should be used to get the total EOS table, GETNUC should be used to get the nuclear EOS table, and then SUBNUC may be used to obtain the electronic EOS. The electronic EOS is obtained by subtracting the nuclear contribution from the total EOS. SUBNUC replaces the total EOS table by the electronic EOS table. If SUBNUC finds no nuclear table available, the Cowan nuclear model† is used to perform the subtraction. Note that SUBNUC assumes for all tables involved that density is in grams per cubic centimeter, temperature is in kiloelectron volts, pressure is in 10⁹ Joules per cubic centimeter, and energy is in 10⁹ Joules per cubic centimeter (not per unit mass). GETEOS, GETNUC, and SUBNUC must all be modified to get tables in other units. SUBNUC has the following calling sequence.

CALL SUBNUC(IR,MID,IDTE,IDTN, TBLS,IFLG)

IR(INPUT) - Material region number.
MID(INPUT) - Sesame material identification number (id).
IDTE(INPUT) - Data-type indicator for electronic EOS: a total EOS table must be loaded under this data type before use by SUBNUC.
IDTN(INPUT) - Data-type indicator for nuclear EOS table.
TBLS(INPUT/ OUTPUT) - Table-storage array (LCM).
IFLG(OUTPUT) = 0 if total EOS table is missing, = 1 if nuclear table is subtracted, and = 2 if Cowan model is subtracted.

SUBNUC assumes the availability of COMMON/PAUL/AMASS(20) where AMASS elements contain the atomic mass and the atomic number double-packed (in that order) for each region. Thus, the atomic mass for region five is stored in the first 30 bits of AMASS(5) and the atomic number is stored in the last 30 bits.

C-36
C. Initial-State Computation Subroutines

The following subroutines may be used for initial-state computation from inverted EOS tables. Although the main function of these routines is initial-state computation, various applications may find other uses for them. Because both routines use an interpolation package numerous times causing execution time to be slow. Hence, these routines are not recommended for extensive use. Note that if scaling, ramps, and special-phase transitions are not required by the application, the calls to T4EOSA in T4RTPE and T4PTRE may be replaced by calls to T4DATI (see Sec. III).

1. T4RTPE. This subroutine computes \( P(\rho, T) \) and \( E(\rho, T) \) and has the following calling sequence.

\[
\text{CALL T4RTPE(IR, IDT, TBLS, R, T, P, E, IFLG)}
\]

- IR(INPUT) - Material region number.
- IDT(INPUT) - Data-type indicator for inverted EOS table.
- TBLS(INPUT) - LCM array where Sesame tables are stored.
- R(INPUT) - Density.
- T(INPUT) - Temperature.
- P(OUTPUT) - Pressure.
- E(OUTPUT) - Energy.
- IFL - Output flag
  = 1 for success and
  = 0 for failure.

2. T4PTRE. This subroutine computes \( \rho(P, T) \) and \( E(P, T) \) and has the following calling sequence.

\[
\text{CALL T4PTRE (IR, IDT, TBLS, P, T, R, E, IFL)}
\]

- IR(INPUT) - Material region number.
- IDT(INPUT) - Data type for inverted EOS tables.
- TBLS(INPUT) - LCM array where Sesame tables are stored.
- P(INPUT) - Pressure.
- T(INPUT) - Temperature.
- R(OUTPUT) - Density.
- E(OUTPUT) - Energy.
- IFL(OUTPUT) - Flag
  = 1 for success and
  = 0 for failure.

D. Auxiliary Subroutines

The following are auxiliary subroutines used by the HYDSES initialization-phase subroutines. Each subroutine will be presented with a brief statement of its purpose and a description of its calling sequence. The HYDSES user need not be familiar with them.

1. MATCHK. This subroutine checks to see if a data table for a given material has been loaded previously.

\[
\text{CALL MATCHK(MID, NRS, LOC, TBLS, IFLAG)}
\]

- MID (INPUT) - Material id.
- NRS (INPUT) - Number of regions.
- LOC (INPUT) - Array of first-word locations for table storage.
- TBLS (INPUT) - Table-storage array (LCM).
- IFLAG (OUTPUT) - IFLAG = 0 if the material has not been loaded. IFLAG gives the location of the table if it has been loaded.

2. TABFCH. This subroutine is used to fetch tables from the Sesame library using random I/O.

\[
\text{CALL TABFCH(MID, TID, LIB, A, LEN, IFLAG)}
\]

- MID (INPUT) - Material id.
- TID (INPUT) - Floating-point table id.
- LIB (INPUT) - Library unit number.
- A (OUTPUT) - LCM array for table storage.
- LEN (INPUT) - Length of A.
- IFLAG (OUTPUT) - Flag
  = 0 for table not found,
  > 0 for number of words in table returned, and
  < 0 for minus the number of extra words needed to store the table.

If TID = 0.0, the material index is returned. See Appendix A for a more detailed discussion.
3. INBUFR. This subroutine is used to perform the random read operation.

CALL INBUFR(LU,Z,NW,IAD,IFLG)

LU(OUTPUT) - Library unit number.
Z(OUTPUT) - LCM array where data are to be stored.
NW(OUTPUT) - Number of words to be read.
IAD(OUTPUT) - Starting disk address of data to be read.
IFLG(OUTPUT) - Flag
= 0 if end of file is encountered,
= 1 if data are read in normal order,
= -1 if error is encountered.

Note: This version of INBUFR uses the LASL/LTSS/FTN random I/O routine RDISK. Thus, INBUFR needs to be modified for use in other systems.

4. DPACK. This function is used to double-pack floating-point numbers into one word.

DPACK(A,B)

A(INPUT) - Floating-point number.
B(INPUT) - Floating-point number.
DPACK(OUTPUT) - Word packed with A in first 30 bits, and B in last 30 bits.

Note that DPACK assumes the use of 60-bit computer words and uses the system-dependent SHIFT function.

5. IONEOS. This subroutine is used to compute the ionic contribution to the equation of state. See Ref. 6 for a discussion of its use.

6. ISRCH. This function is used to find an index associated with a value in an array.

ISRCH(X,TBLS,N)

X(INPUT) - Value to be located.
TBLS(INPUT) - Array to be searched (LCM).
N(INPUT) - Number of values to be searched.
ISRCH(OUTPUT) - = I if TBLS(I) ≤ X < TBLS(I+1) and

Note that values in TBLS must be contiguous and in ascending order. A more flexible search routine (ISRCHK) is discussed next.

7. ISRCHK. This function is used to find an index associated with a value in an array.

ISRCHK(X,TBLS,N,K,NSFT)

X(INPUT) - Value to be located.
TBLS(INPUT) - LCM array to be searched.
N(INPUT) - Number of values to be searched.
K(INPUT) - Spacing between values.
NSFT(INPUT) - Number-of-bits array values are to be shifted.

Note that array TBLS may be either in ascending or descending order; array values need not be contiguous, and double-packed words may be interrogated.

8. INV301. This subroutine is used in subroutine GETINV to convert a temperature-based EOS table into an energy-based EOS table.

CALL INV301(DSTR,LOC,RO,LDS)

DSTR(INPUT) - LCM table-storage array.
LOC(INPUT) - Location of Sesame-type 301 table in DSTR.
RO(INPUT) - Approximate solid density.
LDS(OUTPUT) - Length of inverted table.

Note that the inverted table replaces the 301 table.

9. T4DAT, T4INTP, RATFN1, T4DATI, and T4EOSA. Although these subroutines are involved in the initialization phase, the major use of these routines is in the computational phase. These computational-phase routines are discussed in the next section. Table VI summarizes their use in the initialization phase.

10. TABSEQ. This subroutine is used to fetch tables from the Sesame library using sequential I/O. The TABSEQ calling sequence is identical to that of TABFCH discussed in Sec. II.D.2. See Appendix A for more details.
### TABLE VI

**USE OF COMPUTATIONAL-PHASE SUBROUTINES IN THE INITIALIZATION PHASE**

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Use in Initialization Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>T4DAT</td>
<td>Called by SUBNUC to interpolate nuclear EOS to total EOS grid</td>
</tr>
<tr>
<td>T4EOSA</td>
<td>Used by T4RTPE and T4PTRE for initial-state computation</td>
</tr>
<tr>
<td>T4DATI</td>
<td>Called by T4EOSA</td>
</tr>
<tr>
<td>T4INTP</td>
<td>Called by T4DAT and T4DATI</td>
</tr>
<tr>
<td>RATFN1</td>
<td>Used by INV301 to invert temperature-based tables; also called by T4DATI</td>
</tr>
</tbody>
</table>

11. **INBUF.** This subroutine is used to perform the binary sequential read operation.

CALL INBUF (LU,Z,N,IM)

- **LU** (INPUT) - Library unit number.
- **Z** (OUTPUT) - LCM array where data are stored.
- **N** (INPUT) - Number of words to be read.
- **IM** (OUTPUT) - Flag

INBUF uses the FORTRAN binary read and end-of-file test. The end-of-file test varies from system to system. Hence, INBUF may need to be modified for use in other systems.
III. COMPUTATIONAL-PHASE
SUBROUTINES

The computational-phase subroutines, T4DAT, T4DATI, and T4EOSA are intended for direct use by the application program. Subroutines RATFN1 and T4INTP are interpolation routines. T4DAT calls the two-dimensional interpolator (T4INTP) to perform temperature-based interpolations, whereas T4DATI uses both the one-dimensional interpolator RATFN1, and T4INTP for inverted EOS computation. T4EOSA is optional and is included to add a higher level of capability when inverted EOS tables are used. Each of these routines has no calling sequence, therefore all parameters are passed in common blocks. The detailed descriptions of the subroutines (given below) include their purpose and common block I/O descriptions. Note that all the computational-phase routines assume that the LCM table-storage array TBLS appears in COMMON/SESOUT/TBLS$(TDIMS$), where $TDIMS$ is the dimension of TBLS.

A. T4DAT

Subroutine T4DAT performs interpolation on temperature-based tables. Input is passed through common block /SESIN/ and output through /SESOUT/.

COMMON/SESIN/IR,IDT,X,Y,IBR,IFL

| IR | Material region number. |
| IDT | Data-type indicator. |
| X | Density. |
| Y | Temperature. |
| IBR | Flag |
| IFL | not used. |

Z1(1) = first dependent variable Z1 = Z1 (x,y),
Z1(2) = $\partial Z1/\partial x$,
Z1(3) = $\partial Z1/\partial y$,
Z2(1) = second dependent variable Z2 = Z2(x,y),
Z2(2) = $\partial Z2/\partial x$, and
Z2(3) = $\partial Z2/\partial y$.

The region and data-type numbers must be valid, that is, data tables must be loaded for the given IR and IDT. Table III lists the available dependent variables. Electronic pressure and energy($P_e$, $E_e$) are available if SUBNUC is used. For certain tables (see Table III), $X = \log_{10}(\rho)$ and $Y = \log_{10}(T)$ should be input, and $\log_{10}\kappa$, $\log_{10}\kappa_p$, $\log_{10}\kappa_r$, $\log_{10}N_T$, and $\log_{10}\kappa_T$ are returned.

B. T4DATI

Subroutine T4DATI performs interpolation on inverted equation-of-state tables. Input is passed through common block /SESIN/ and output through /SESOUT/.

COMMON/SESIN/IR,IDT,X,Y,IBR,IFL

| IR | Material region number. |
| IDT | Data-type indicator for inverted tables. |
| X | Density. |
| Y | Internal energy. |
| IBR | Flag |
| IFL | not used. |

COMMON/SESOUT/Z1(3), Z2(3)

Z1(1) = P(x,y),
Z1(2) = $\partial P/\partial x$,
Z1(3) = $\partial P/\partial y$,
Z2(1) = T(x,y),
Z2(2) = $\partial T/\partial x$, and
Z3(3) = $\partial T/\partial y$. 

C-40
C. T4EOSA

Subroutine T4EOSA is optional and provides an added level of capability over T4DATI.

T4EOSA requires as input the /SESIN/ block as described for T4DATI (except here the IFL flag has meaning) and the following common block.

COMMON/EOSCOM/SR($NREG$), ES ($NREG$), RO($NREG$), A1($NREG$), A2($NREG$), A3($NREG$), EM($NREG$).

Here, $NREG$ is the dimension used for regional arrays. SR, ES, RO, A1, A2, A3, and EM are supplied on a region basis and are discussed below in conjunction with the features provided by T4EOSA. Output from T4EOSA is returned in /SESOUT/ as discussed for T4DATI. Because of the scaling features of T4EOSA, the values of x and y (in /SESIN/) upon exit may differ from those values when the routine was entered. T4EOSA is used to provide the following features for inverted EOS computation.

(1) Density and energy scaling.

\[ p (p,E) = P_{TAB}(\rho T,E T) \],

\[ \rho T = (SR)p, \text{ and} \]

\[ E T = (E + ES)/SR, \]

where \( P, \rho, \text{ and } E \) are the pressure, density, and energy variables used by the code, and \( P_{TAB}(\rho T,E T) \) is the tabular EOS.

The parameter SR is useful for treating isotopic mixtures. If \( A T \) is the atomic mass for the EOS table, an EOS for an atomic mass \( A \) is obtained by setting

\[ SR = A T/A . \]

For example, set \( SR = 2 \) to scale the Sesame D4 EOS, #5263, to H4. Similarly, set \( SR = 0.80 \) to obtain an EOS for a 50:50 DT mixture.

The parameter ES can be used to change the energy zero of the table. It is intended for use primarily with the "ramp" option, discussed in (2).

(2) Foams and phase transitions.

A1, A2, A3 = ramp parameters,

ES = energy shift in user units,

EM = "melt" energy in user units, and

RO = \( \rho o \) = initial density.

For treatment of foams and certain types of phase transitions, it is possible to modify the Sesame EOS by adding a "ramp" that describes the behavior of the material at low stress. This ramp is shown in Fig. 1.

The material starts out in either a porous state or low-density phase. The EOS is given by

\[ P = A1 \left( \rho / \rho o - 1 \right) , \]

where \( \rho o \) is the initial density and \( A1 \) is the bulk modulus. \( A1 \) can be computed from

\[ A1 = \rho o C_o^2 \text{ (GPa)} , \]

where \( C_o \) is the bulk sound speed (in km/s). If \( A1 = 0 \), no ramp calculation is performed.

At some pressure \( P_c \), the material will begin to "crush," or transform to the high-density phase.
described by the Sesame EOS. The EOS of the crush curve is

\[ P = A_2 \left( \rho / \rho_o - A_3 \right) \]

(If \( A_2 = 0 \), there is no crush region and Eq. (1) is continued on until it crosses the Sesame hydrostat.) \( A_2 \) is related to the transition pressure by

\[ A_2 = \frac{P, A_1}{\frac{P, A_1}{1 + A_1(1 - A_3)}} \quad \text{(user units)} \]

The transition pressure for foams is usually rather small (< 1 kbar). For a phase transition, \( P \), must be obtained from experiment. The parameter \( A_3 \) can be adjusted to give the correct slope of the crush curve. In the absence of data, the default value (\( A_3 = 0 \)) should give acceptable results.

At some pressure \( P_i \), the ramp crosses the Sesame hydrostat. At that point, the substance is said to be "crushed." Subsequently, the material may behave either reversibly (follow the ramp on expansion) or irreversibly (stay in the high-density phase on expansion). Foams normally are irreversible, but phase transitions may be either. The behavior is controlled by setting the IFL flag appropriately. If \( IFL = 0 \) before \( T4EOSA \) is entered, the code will choose between the ramp and the tables for EOS values. It will return \( IFL = 0 \) if the ramp was used, and \( IFL = 1 \) if the tables were used. If \( IFL = 1 \) before \( T4EOSA \) is entered, the code will always go to the tables for the EOS. Thus, if \( IFL = 0 \) upon each entry to \( T4EOSA \), the transition is reversible. If \( IFL = 0 \) initially and not reset by the user's code, then the transition is irreversible. The material will also behave irreversibly if it melts, that is, if the melt energy is exceeded. \( EM \) depends on the path and must be treated carefully. If \( EM \) is sufficiently high, "melting" will not occur in most cases of interest.

The energy shift is defined by

\[ ES = -\Delta E \left( \text{initial} - \text{final} \right) \]

where \( \Delta E \) is the energy required to transform the low-density phase to the high-density phase. Hence, there are two cases. If the initial phase is stable, \( ES \) is negative. If the initial phase is metastable, \( ES \) is positive.

(3) Recommended default values.

Recommended default values for parameters appearing in /EOSCOM/ are

- \( SR = 1.0 \)
- \( ES = 0 \)
- \( RO = \text{normal solid density (g/cm}^3\text{)} \)
- \( A1 = 0 \)
- \( A2 = 0 \)
- \( A3 = 0 \), and
- \( EM = 1000.0 \text{ Mbar} \cdot \text{cm}^4/\text{g} \).

D. RATFN1

Subroutine RATFN1 is the one-dimensional interpolation routine. RATFN1 communicates with other subroutines using the following common blocks.

COMMON/INTORD/IFN
COMMON/RTBLK1/LOCX,KX,LOCY,KY,I,N,IP,X,Y(2)

where

IFN(INPUT) - Integration type
  IFN = 0 for rational function interpolation and
  IFN = 1 for linear interpolation

LOCX(INPUT) - Location of \( x \) vector in array TBLS.

KX(INPUT) - Spacing of \( x \) vector.

LOCY(INPUT) - Location of \( y \) vector in TBLS.

KY(INPUT) - Spacing of \( y \) vector.
  Index into \( x \) and \( y \) vector corresponding to requested \( x \) value.

N(INPUT) - Length of \( x \) and \( y \) vectors.

IP(INPUT) - Flag
  = 0 if coefficients from previous call are to be used
  = 1 if coefficients are to be recalculated.

X(INPUT) - Value of \( x \) for which \( y \) is requested.

Y(OUTPUT) - \( Y(1) = y(x) \) and
  \( Y(2) = \partial y/\partial x \).
E. T4INTP

Subroutine T4INTP is the two-dimensional interpolation subroutine. T4INTP communicates with subroutines calling it through the following common blocks.

COMMON/INTORD/IFN
COMMON/RTBLK2/LOCX,IX,NX,LOCY, IY,NY,LOCZ,NZ,NSFT,X,Y,Z(3),IP,IDS, ZZ($CDIM$)

IFN(INPUT) - = 0 for rational function interpolation and
- = 1 for bilinear interpolation.

LOCX(INPUT) - Location of the x vector in array TBLS.

IX(INPUT) - Index into x vector corresponding to x value.

NX(INPUT) - Length of x vector.

LOCY(INPUT) - Location of the y vector in array TBLS.

IY(INPUT) - Index into y vector corresponding to y value.

NY(INPUT) - Length of y vector.

LOCZ(INPUT) - Location of dependent variable array Z(x,y) in array TBLS.

NZ(INPUT) - Spacing of values in Z array.

NSFT(INPUT) - Number of bits that dependent variables are shifted when accessed.

X(INPUT) - Independent variable, x-value.

Y(INPUT) - Independent variable, y-value.

Z(OUTPUT) - Array of length 3 where
- Z(1) = Z(x,y),
- Z(2) = ∂Z/∂x, and
- Z(3) = ∂Z/∂y.

IP(INPUT) - Flag
- = 0 use previous interpolation coefficients and
- = 1 recompute interpolation coefficients.

IDS(INPUT) - Index into array ZZ where coefficients are to be stored or located.

ZZ(INPUT) - Interpolation coefficient array of dimension $CDIM$.
IV. GENERAL USE OF HYDSES

This section describes how to install HYDSES into an applications program.

A. Preliminaries

The user should determine
(1) the maximum number of regions ($\$NREGS$) needed by the application for dimensioning purposes,
(2) the maximum number of Sesame data types ($\$NDATS$) needed by the application for dimensioning purposes,
(3) the amount of LCM table storage required for dimensioning purposes. (The user should allow approximately 3000 words for an EOS table and 1550 words for an opacity table.)
(4) the interpolation type to be used (see Sec. III.D, III.E).

B. Specifications

The following must be specified in the user's code before HYDSES subroutines are called.

(1) The Sesame (in-core) directory common block

\[ \text{COMMON/S2DIR/LCMX,NRS,LCFW} \]
\[ ($\$NREGS$,\$NDATS$) \]

Set LCMX to the number of words available for table storage; NRS to the number of regions in the problem ($\leq$ $\$NREGS$), and LCFW($\$NREGS$, \$NDATS$) to all zeroes. After initialization, LCFW will contain the first-word location in the storage array of the table to be used for each region and data type.

(2) The Sesame LCM common block for table storage

\[ \text{COMMON/SESDAT/TBLS($\$TDIMS$)} \]

(3) The interpolation-type common block

\[ \text{COMMON/INTORD/IFN} \]

Note that changing IFN during the computational phase may cause problems. Set IFN as discussed in Secs. III.D and III.E.

(4) Initialize the table-storage-pointer word (LCNT) to 1.

(5) Assign unit numbers to the Sesame data files to be used.

C. Replacements and Selection of Subroutines

The user should then replace (using a text editor) all occurrences of $\$NREGS$, $\$NDATS$, and $\$TDIMS$ in HYDSES by those chosen in Sec. IV.A. Also, $\$TDIMS$ should be replaced with the product of $\$NDATS$ and 32. The user should then determine which of the HYDSES subroutines are needed. The "get" routines should be checked to see if they yield tables in the appropriate units (see Tables III and IV).

D. Initialization Phase

Now, using the subroutines described in Sec. II, the user may code the initialization phase appropriate to the application.

E. Computational Phase

After initialization is complete, the routines described in Sec. III may be used for computational-phase coding.
V. TEST EXAMPLE

The following example (Fig. 2) illustrates the use of the HYDSES package. The code used to generate the example is shown in Appendix B. The purpose of the code is to compute $P$, $E$, $r_n$, $r_p$, $r_e$, and $N_e$ as functions of $\rho$ and $T$. The user is first prompted for the number of regions and a set of associated material numbers. After these have been entered, the program enters an initialization phase where the desired Sesame tables are loaded from the disk libraries for each region. In the example that follows, three regions are loaded with data; the first with deuterium, the second with aluminum, and the third with iron. After the tables are loaded, the program enters a "mini" computational phase. Here the user is prompted for region number, density, and temperature. Note that the units correspond to those in Table III. After these values are entered, the dependent variables are computed in the same manner they would be in an applications program. After the program prints out the dependent variables at the terminal, the user is prompted for further values of region, density, and temperature. This sequence continues until the user types in 0 0 0.
HYDITX / 1 1 .6
ENTER THE NUMBER OF REGIONS
? 3
ENTER SESAME EOS MATERIAL NUMBERS
? 5263 3710 2140
ENTER REGION, DENSITY, TEMPERATURE
? 1 8 .5 .7
PRESSURE = 5.7322E-01
ENERGY = 8.6419E-01
PLNK OP = 2.8531E+00
TH COND = 1.3743E-05
NO FR E = 1.0000E+00
ENTER REGION, DENSITY, TEMPERATURE
? 2 8 .5 .7
PRESSURE = 2.7150E-01
ENERGY = 5.0821E-01
PLNK OP = 6.5492E+02
TH COND = 4.8022E-06
NO FR E = 1.2248E+01
ENTER REGION, DENSITY, TEMPERATURE
? 3 8 .5 .7
PRESSURE = 2.2553E-01
ENERGY = 4.9409E-01
PLNK OP = 3.9797E+02
TH COND = 3.4905E-06
NO FR E = 2.2048E+01
ENTER REGION, DENSITY, TEMPERATURE
? 1 4 .25 .35
PRESSURE = 1.4288E-01
ENERGY = 2.1662E-01
PLNK OP = 3.0243E+00
TH COND = 2.8165E-06
NO FR E = 1.0000E+00
ENTER REGION, DENSITY, TEMPERATURE
? 2 4 .25 .35
PRESSURE = 6.1422E-02
ENERGY = 1.2676E-01
PLNK OP = 8.4304E+02
TH COND = 1.1244E-06
NO FR E = 1.0854E+01
ENTER REGION, DENSITY, TEMPERATURE
? 3 4 .25 .35
PRESSURE = 4.6630E-02
ENERGY = 1.1895E-01
PLNK OP = 3.0076E+03
TH COND = 9.0489E-07
NO FR E = 1.8891E+01
ENTER REGION, DENSITY, TEMPERATURE
? 0 0 0
STOP FTN
HYDITX LATSS TIME 3.691 SECONDS
CPU= .228 sys= .024 i/o= 3.444
ALL DONE

Fig. 2. Test example.
VI. PROGRAM AVAILABILITY

HYDSES may be obtained from the Common File System (CFS) for LASL users by using the LTSS execute line

MASS GET /088077/HYD/HYDSES.

Non-LASL users may obtain HYDSES by contacting

Sesame Library, MS 925
Los Alamos Scientific Laboratory
P.O. Box 1663
Los Alamos, NM 87545

and providing a magnetic tape on which the HYDSES source code will be written.

REFERENCES


APPENDIX A

SESAME LIBRARY STRUCTURE AND RANDOM I/O

A Sesame library consists of a directory file followed by a variable number of material files (see Fig. A-1). Each material file consists of an index record followed by a variable number of data records. The index record format is presented in Table A-I. The data records include the tables described in Sec. II. The directory file consists of two records. These records are described in Table A-II.

The address table appearing in the second record of the directory file supplies addresses that can be used to locate the beginning of each material file. This address table is generated by a new and improved version of the Sesame update package. This package will be sent to users along with the HYDSES package upon request. The addresses in this table are computed assuming that each record is terminated by a one-word record mark and each file is terminated by a one-word file mark. The address computation is performed in subroutines S2BLDL before statement 10. This subroutine must be modified if a different addressing scheme is to be used. Note that the library generated by this package can be read with both sequential and random I/O on LASL/LTSS computers. On other systems, the use of a postprocessing subroutine may be necessary to write the library randomly if the random I/O option is desired.

Subroutine TABFCH (see Sec. II) is used to fetch specified data tables from the Sesame library using random I/O. TABFCH uses the random I/O read subroutine INBUFR. The first word of the directory file is read. This is used to compute the length of the second record of the directory file. This record is then read in starting at address 5. The address of the material index record and its length are obtained from the second record of the directory file for the requested material. The index record is read randomly using the obtained address and length. The address for the requested data table is computed from the starting address of the material file and from length information stored in the index record. The data table is obtained using the address. This subroutine must be modified to implement other addressing schemes.

Subroutine TABSEQ is included in the HYDSES package for those users who wish to avoid system-dependent random I/O. TABSEQ is a sequential I/O version of TABFCH. The TABSEQ calling sequence is identical to that of TABFCH. TABSEQ calls the sequential binary read subroutine INBUF (instead of INBUFR). Use of the address table is avoided by TABSEQ. To use TABSEQ in HYDSES, all the user must do is replace calls to TABFCH by calls to TABSEQ. Note that use of sequential I/O (TABSEQ) is much slower than random I/O (TABFCH). Installations that already support a Sesame library may use TABSEQ to avoid the regeneration of the library (using the new update package) to include the address table.

Fig. A-1.
Structure of the Sesame data library.
TABLE A-I

STRUCTURE OF THE INDEX RECORD FOR A SESAME MATERIAL FILE

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATID</td>
<td>Material id number (real number).</td>
</tr>
<tr>
<td>DATE1</td>
<td>Date of creation of material.</td>
</tr>
<tr>
<td>DATE2</td>
<td>Date when material file was last updated.</td>
</tr>
<tr>
<td>VERS</td>
<td>Version number of most recent update.</td>
</tr>
<tr>
<td>NREC</td>
<td>Number of data records (real number).</td>
</tr>
<tr>
<td>TBLID</td>
<td>Table of catalog numbers for the data records, in the same order as they are stored in the material file.</td>
</tr>
<tr>
<td>NWDS</td>
<td>For each TBLID(I), NWDS(I) is the number of words in the data record (real number).</td>
</tr>
</tbody>
</table>

TABLE A-II

STRUCTURE OF THE DIRECTORY FILE FOR THE SESAME LIBRARY

Record 1. Consists of the three words N, DATE, and VERS.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Number of material files in the library (real number).</td>
</tr>
<tr>
<td>DATE</td>
<td>Date of current version.</td>
</tr>
<tr>
<td>VERS</td>
<td>Version number.</td>
</tr>
</tbody>
</table>

Record 2. Consists of the three arrays (MATID(I), I = 1, N), (NWDS(I), I = 1, N), and (IADR(I), I = 1, N).

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATID</td>
<td>Table of material id numbers, in the same order as the material files in the library (real numbers).</td>
</tr>
<tr>
<td>NWDS</td>
<td>For each material file MATID(I), NWDS(I) is the number of words in the index record (real numbers).</td>
</tr>
<tr>
<td>IADR</td>
<td>IADR(I) is the address of each index record (real numbers).</td>
</tr>
</tbody>
</table>
APPENDIX B

CODE USED TO GENERATE THE TEST EXAMPLE

LASL Identification No. LP-2047

1 $FTN (I=:ME), CNAME=:ME, L=LIST, GLIS=HYBBIN
2 PROGRAM TEST (TTY, TAPE59=TTY)
3 C .. HYBBIN IS THE BINARY RELOCATABLE CORRESPONDING THE HYDSES
4 C .. SOURCE CODE
5 C .. LEVEL 2, T5LS
6 C .. DIMENSION ZB(3), MAT(10)
7 C .. DIRECTORY COMMON BLOCK
8 COMMON/S2DIR/LCMX,NAS,LCFW(20,5)
9 C .. TABLE STORAGE AREA
10 COMMON/SESDAT/T5LS(25000)
11 C .. INTERPOLATION TYPE COMMON
12 COMMON/INTPD/IFN
13 C .. COMMNS TO COMMUNICATE WITH COMPUTATIONAL PHASE SUBROUTINES
14 COMMON/SESIN/IR,IDT,XY,IBR,IIFL
15 COMMON/SESDOUT/Z1(3),Z2(3)
16 C .. INSURE THAT THE LCFW ARRAY IS ZERODED
17 DATA LCFW/100*0/
18 LTNT=59
19 C .. SET FOR RATIONAL FUNCTION INTERPOLATION
20 IFN=0
21 C .. SET DIMENSION OF T5LS
22 LCMX=25000
23 C .. SET TABLE POINTER WORD
24 LCNT=1
25 C .. ASSIGN EOS LIBRARY
26 LES=1
27 CALL GASSIGN (LES,5HSES2L,0,0)
28 C .. ASSIGN OPACITY LIBRARY
29 LOP=2
30 CALL GASSIGN (LDP,6HSES2OP,0,0)
31 WRITE (LTNT,9000)
32 9000 FORMAT(" ENTER THE NUMBER OF REGIONS")
33 READ (LTNT,*), NAS
34 WRITE (LTNT,9010)
35 9010 FORMAT(" ENTER SESAME EOS MATERIAL NUMBERS")
36 READ (LTNT,*), (MAT(J), J=1,NAS)
37 C .. START INITIALIZATION PHASE
38 DO 100 J=1,NAS
39 MID=MAT(J)
40 C .. GET EOS TABLE FOR THIS REGION AND ASSIGN DATA TYPE = 1
APPENDIX B (cont)

41 C FOR EOS DATA
CALL GETEOS(J,MID,1,TBSL,LCNT,LES,IFL,Z)
44 C . GET THE ROSSLAND/PLANCK OPACITY TABLE AND ASSIGN IDT=2
INCREMENT MID BY 10000 TO GET OPACITY NO.
MID=MID+10000
CALL GETPO(J,MID,2,TBSL,LCNT,LOD,IFL)
47 C . GET THE THERMAL CONDUCTIVITY TABLE AND ASSIGN IDT=3
CALL GETTCE(J,MID,3,TBSL,LCNT,LOD,IFL)
49 C . GET THE NUMBER OF FREE ELECTRONS TABLE AND ASSIGN IDT=4
CALL GETNFE(J,MID,4,TBSL,LCNT,LOD,IFL)
60 C CONTINUE
200 WRITE(LTTY,9020)
9020 FORMAT(" ENTER REGION DENSITY TEMPERATURE")
READ(LTTY,*) IF0XY IF(IF0,0) STOP
C . COMPUTE EOS
IDT=1
200 IDT=0
CALL T4DAT
P=21.1
E=22.1
C . COMPUTE THE ROSSLAND AND PLANCK OPACITY
IDT=2
X=ALOG10(X)
Y=ALOG10(Y)
CALL T4DAT
POF=10.0**Z1(1)
POF=10.0**Z2(1)
C . COMPUTE THE THERMAL CONDUCTIVITY
IDT=3
200 IDT=1
CALL T4DAT
VE=10.0**Z1(1)
C . COMPUTE THE NUMBER OF FREE ELECTRONS
IDT=4
CALL T4DAT
FNP=10.0**Z1(1)
C . WRITE OUTPUT TO TTY
80 WRITE(LTTY,9050) P,E,POF,POF,VE,FNP
9050 FORMAT(1X,10WPRESSURE ,2H= ,1PE12.4,/
$1X,10WENERGY ,2H= ,E12.4,/
$1X,10WROSS OP ,2H= ,E12.4,/
$1X,10WPLNK OP ,2H= ,E12.4,/
$1X,10WTH COND ,2H= ,E12.4,/
$1X,10WHNO FR E ,2H= ,E12.4)
END
INSTRUCTIONS FOR USE OF THE SESAME LIBRARY

The Sesame library is a collection of material properties tables which are kept on a disk file, magnetic tape, or other permanent mass storage device. The library can be accessed by computer programs and used to compute various quantities of interest to users. These instructions explain how to establish, update, and use a copy of the Sesame library.

You have received a computer listing and two or more magnetic tapes. Tapes marked SES01, SES02, etc., contain numerical data from the Sesame library; the tape marked SES2I contains the library of computer subroutines given in the listing. These tapes are written in an 80 character card image format so that they can be read and interpreted by your computing machines. For efficient storage and data acquisition, the data tapes must be preprocessed to produce a binary-formatted file (or used to update an existing file) on your local computing system. This procedure, described below, uses subroutines from the tape SES2I.

The parameters used in writing the tapes should be compatible with your computer. The tapes may be 7-track or 9-track, 800 or 1600 B.P.I., and the character set may be either BCD or EBCDIC. (On EBCDIC tapes, each record contains 660 bits. Only the first 640 bits, 80 characters, contain data; the remaining 20 bits are blank.) These parameters are noted on the tapes themselves.

Feel free to contact us if you encounter problems or want additional information.

The Subroutine Library - System Adaptation

The file SES2I contains FORTRAN subroutines which are needed to preprocess the data tape, access data from the binary library, and compute various functions by search and interpolation. We have attempted to make these routines easy to use, compiler independent, and modular, and to provide comment cards which describe the procedures used. However, there are two problems which the user should consider.

1. Binary Input/Output: All binary I/O is confined to the two routines, INBUF and OUTBUF. Users may choose to substitute more efficient commands for the unformatted READ/WRITE statements. For example, BUFFER IN/BUFFER OUT is usually preferred on CDC machines.

These routines also use an end of file (EOF) write and an EOF test. There exist no standard FORTRAN commands for these procedures. Moreover, these methods are not suitable on some systems, including IBM. In such cases, the system EOF should be replaced by writing (and testing for) a special Hollerith string, such as 2HFM. Versions of INBUF and OUTBUF routines which will work on IBM systems are listed at the end of these instructions.

2. Use of Large Core Memory: When accessing and using data from the library, an array for storage of the tables must be declared in a DIMENSION or COMMON statement. The name of this array is passed in the argument list of the Sesame subroutines, as discussed below. On CDC 7600 machines, a user can easily adapt the routines so that this storage is in large core memory (LCM). In
order to avoid compiler-dependent features, we have used the simplest, but
least efficient approach -- directly addressed LCM. However, the running
time of the routines increases by only 5-10% in the LCM mode, so that there
is little to gain by going to more complicated procedures.

In each routine, the arrays to be declared LCM are designated by a comment
card, with C**** in columns 1-5. The user must replace this card with a
declaration appropriate to his compiler (i.e., LARGE, LCM, etc.).

Preprocessing the Data Tape - Subroutine UPDATE

Subroutine UPDATE is used either to create or to make changes in a binary
library file from a BCD data tape. As input to this routine the user must
supply the BCD tape (LUP), the library file (LIB), a scratch file (LM), and a
file on which the new library (created or updated) will be written (LNEW). If
the user specifies the creation of a new library (INEW=0), the old library file
(LIB) is not used. If the user specifies an update to an existing library
(INEW=1), the old library remains intact after the update. Usage is as follows:

CALL UPDATE (DATE, NCPW, FMT, INEW, LUP, LIB, LM, LNEW, LP)

DATE - Date of updating library (a six digit real number)

NCPW - Number of characters per word

FMT - Format for reading Hollerith data

INEW - 0 (create a new library),

   1 (update a existing library)

LUP - Unit number of BCD update file

LIB - Unit number of original library

LM - Unit number of scratch file

LNEW - unit number of new (created/updated) library

LP - Unit number of printer file for diagnostic messages

Because different computing machines have different word lengths for Hollerith
data, the user must supply the appropriate format for reading 80 BCD characters.
For example, CDC machines permit up to 10 BCD characters per word, while IBM
machines allow a maximum of 4. In these two cases, the appropriate parameters
would be as follows.

CDC:  NCPW = 10, FMT = 6H (8A10)

IBM:  NCPW = 4, FMT = 6H (20A4)

The UPDATE routine will operate most efficiently if BCD unit, LN, is a disk file
rather than a tape.

Accessing Data from the Library

Each material in the Sesame library is identified by a 4-digit material
I.D. number from 1000 to 9999. For each I.D. number, the library contains a
material file which consists of several data records. At present, the library supplies only equation of state information in the material files. Therefore, this discussion will be limited to the computation of thermodynamic functions. In the future, however, opacity tables and other kinds of material data will be added to the library.

The routines S2GET and S2EOS are used to compute pressure and internal energy as functions of density and temperature.

\[
P = F_1(\rho, T),
\]

\[
E = F_2(\rho, T),
\]

These programs are particularly useful for hydrodynamic codes. They provide for computations involving several Sesame tables, for the use of Sesame tables along with other EOS options, and for the specification of the same Sesame table in more than one region. All the necessary bookkeeping is internal to the routines.

The user must allocate an array for storage of the EOS tables (storage requirements are discussed below) and provide a COMMON block for a directory to this array. The COMMON block has the form

```
COMMON/S2DIR/LCMX, NRS, LCFW (10),
LCMX - length of the array,
NRS - number of material regions (see below),
LCFW - an array used as a directory, initialized to zero by the user.
```

Usage of the routine S2GET is as follows.

```
CALL S2GET (IR, IDS2, TBLS, LCNT, LU, IFL),
IR - material region number,
IDS2 - Sesame material number,
TBLS - name of array designated for storage of tables,
LCNT - current word in array TBLS,
LU - unit number for library,
IFL - error flag.
```

Each "region" corresponds to a different material. Regions should be numbered consecutively, whether or not a Sesame EOS is used. The same EOS may be requested for more than one region -- the routine will load a given table only once. S2GET must be called once for each region for which a Sesame EOS is required.
Data is loaded into the array TBLS, beginning with the word TBLS (LCNT). After successful execution of the routine, LCNT is set to the first word location following the data string which has been loaded. S2GET can be called again, using this new value of LCNT. However, the user can load other data into the array and compute a new value of LCNT before calling S2GET again.

If the routine is successful, IFL = 1 is returned. If IFL = 0, S2GET was unable to locate the EOS tables on the file LU. If IFL < 0, there is insufficient storage in the array TBLS, and the size of the array should be increased by at least \(|IFL|\) words. This feature will enable some users to allocate storage dynamically.

The routine S2EOS computes pressure, internal energy, and their derivatives, as function of density and temperature. Usage is as follows.

\[
\text{CALL S2EOS (IR, TBLS, R, T, P, E),}
\]
\[
\text{IR = material region number,}
\]
\[
\text{TBLS = name of array which contains the EOS tables,}
\]
\[
\text{R = density in Mg/m}^3,
\]
\[
\text{T = temperature in degrees Kelvin,}
\]
\[
\text{P, E = pressure, internal energy vectors,}
\]
\[
P(1), E(1) - \text{pressure in GPA, internal energy in MJ/kg,}
\]
\[
P(2), E(2) - \text{density derivatives, } (\partial P/\partial \rho)_T, (\partial E/\partial \rho)_T,
\]
\[
P(3), E(3) - \text{temperature derivatives, } (\partial P/\partial T)_\rho, (\partial E/\partial T)_\rho.
\]

The parameters IR and TBLS should be identical to those which were used in calling S2GET. Note that the region number, not the Sesame I.D. number, is used to identify the particular EOS table. Hence the user is saved a certain number of bookkeeping chores.

The routine S2GETI and S2EOSI are used to compute pressure and temperature as functions of density and internal energy.

\[
P = G_1(\rho, E),
\]
\[
T = G_2(\rho, E).
\]  \hspace{1cm} (2)

These routines are very similar to those described above. Usage of S2GETI is identical to S2GET. However, S2GETI reformats the EOS tables in order to make the computations with S2EOSI more efficient. Usage of S2EOSI is as follows.

\[
\text{CALL S2EOSI (IR, TBLS, R, E, P, T),}
\]
\[
\text{IR = material region number,}
\]
\[
\text{TBLS = name of array which contains the EOS tables,}
\]

C-55
R = density in Mg/m$^3$,
E = internal energy in MJ/kg,
P, T = pressure, temperature vectors,
P(1), T(1) - pressure in GPA, temperature in Kelvins,
P(2), T(2) - density derivatives, \((\partial P/\partial \rho)_E\), \((\partial T/\partial \rho)_E\),
P(3), T(3) - energy derivatives, \((\partial P/\partial E)_\rho\), \((\partial T/\partial E)_\rho\).

For certain materials, the library also has tables of the pressure, temperature, density, and internal energy along the vapor-liquid coexistence curve. This information is needed in reactor safety problems. Routines S2GET and S2GETI can be modified to access the coexistence data, and routine LA401A can be used to compute the thermodynamic quantities. Users who are interested in reactor safety applications should inquire about the special computer package which has been written for the SIMMER-II code.

Storage Requirements

There is no standard density-temperature grid for Sesame EOS tables. Each grid can be constructed to give the best representation of the EOS with as few points as possible. Consequently, storage requirements vary from material to material. At the present time, most tables require approximately 5000 words. It is our plan to redo most of the tables and reduce this requirement to about 3000 words.

In most cases, a Sesame EOS table covers a much larger density and temperature range than is needed for a particular application. The file SES2I provides a routine, WDW301, which reduces the size of the EOS tables by deleting data outside of specified density and temperature limits. Some users may want to call this routine from within S2GET or S2GETI. We expect that use of this option could decrease storage requirements by an order of magnitude in many cases.

A Test Problem

To test the above procedures and to illustrate use of the Sesame library, we have constructed a test problem. Subroutine TPRB1, which is given in file SES2I, accesses the tables for several materials specified by the user, calculates Hugoniot curves, and prints out the results. Usage is as follows.

CALL TPRB1 (IDS2, N, LU, LP),
IDS2 - An array, of dimension N, giving the material numbers selected by the user,
N - number of materials specified,
LU - unit number for library,
LP - unit number for line printer.
Attached is a sample of the output from this routine, called with the following material numbers.

IDS2(1) - 3200 (lead),
IDS2(2) - 3330 (copper),
IDS2(3) - 3710 (aluminum).

The output variables are as follows.

R - density in Mg/m³,
P - pressure in GPa,
E - internal energy in MJ/kg,
T - temperature in degrees Kelvin,
US - shock velocity in km/sec,
UP - particle velocity in km/sec.
SAMPLE RESULTS FROM SUBROUTINE TPRB1

### SHOCK HUGONIOT FOR SESAME MATERIAL NUMBER 3200

<table>
<thead>
<tr>
<th>P (GPa)</th>
<th>E (MJ/kg)</th>
<th>T (KELVIN)</th>
<th>US (KM/SEC)</th>
<th>UP (KM/SEC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>.158e+02</td>
<td>.400e+02</td>
<td>.500e+00</td>
<td>.152e+04</td>
<td>.353e+01</td>
</tr>
<tr>
<td>.190e+02</td>
<td>.113e+03</td>
<td>.200e+01</td>
<td>.770e+04</td>
<td>.497e+01</td>
</tr>
<tr>
<td>.217e+02</td>
<td>.214e+03</td>
<td>.450e+01</td>
<td>.168e+05</td>
<td>.628e+01</td>
</tr>
<tr>
<td>.244e+02</td>
<td>.341e+03</td>
<td>.800e+01</td>
<td>.278e+05</td>
<td>.753e+01</td>
</tr>
<tr>
<td>.264e+02</td>
<td>.497e+03</td>
<td>.125e+02</td>
<td>.396e+05</td>
<td>.876e+01</td>
</tr>
<tr>
<td>.284e+02</td>
<td>.680e+03</td>
<td>.180e+02</td>
<td>.533e+05</td>
<td>.999e+01</td>
</tr>
<tr>
<td>.301e+02</td>
<td>.391e+03</td>
<td>.245e+02</td>
<td>.687e+05</td>
<td>.112e+02</td>
</tr>
<tr>
<td>.318e+02</td>
<td>.113e+04</td>
<td>.320e+02</td>
<td>.824e+05</td>
<td>.124e+02</td>
</tr>
</tbody>
</table>

### SHOCK HUGONIOT FOR SESAME MATERIAL NUMBER 3330

<table>
<thead>
<tr>
<th>P (GPa)</th>
<th>E (MJ/kg)</th>
<th>T (KELVIN)</th>
<th>US (KM/SEC)</th>
<th>UP (KM/SEC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.093e+01</td>
<td>4.874e+01</td>
<td>5.000e-01</td>
<td>4.437e+02</td>
<td>5.458e+00</td>
</tr>
<tr>
<td>1.251e+01</td>
<td>1.247e+02</td>
<td>2.000e+00</td>
<td>2.071e+03</td>
<td>6.984e+00</td>
</tr>
<tr>
<td>1.385e+01</td>
<td>2.264e+02</td>
<td>4.500e+00</td>
<td>5.375e+03</td>
<td>8.450e+00</td>
</tr>
<tr>
<td>1.503e+01</td>
<td>3.521e+02</td>
<td>8.000e+00</td>
<td>1.027e+04</td>
<td>9.857e+00</td>
</tr>
<tr>
<td>1.620e+01</td>
<td>4.976e+02</td>
<td>1.250e+01</td>
<td>1.589e+04</td>
<td>1.114e+01</td>
</tr>
<tr>
<td>1.730e+01</td>
<td>6.645e+02</td>
<td>1.800e+01</td>
<td>2.236e+04</td>
<td>1.240e+01</td>
</tr>
<tr>
<td>1.834e+01</td>
<td>8.529e+02</td>
<td>2.450e+01</td>
<td>2.979e+04</td>
<td>1.364e+01</td>
</tr>
<tr>
<td>1.932e+01</td>
<td>1.063e+03</td>
<td>3.200e+01</td>
<td>3.304e+04</td>
<td>1.483e+01</td>
</tr>
</tbody>
</table>

### SHOCK HUGONIOT FOR SESAME MATERIAL NUMBER 3710

<table>
<thead>
<tr>
<th>P (GPa)</th>
<th>E (MJ/kg)</th>
<th>T (KELVIN)</th>
<th>US (KM/SEC)</th>
<th>UP (KM/SEC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.182e+00</td>
<td>1.784e+01</td>
<td>5.000e-01</td>
<td>1.713e+02</td>
<td>6.607e+00</td>
</tr>
<tr>
<td>3.615e+00</td>
<td>4.283e+01</td>
<td>2.000e+00</td>
<td>7.205e+02</td>
<td>7.905e+00</td>
</tr>
<tr>
<td>4.006e+00</td>
<td>7.455e+01</td>
<td>4.500e+00</td>
<td>1.971e+03</td>
<td>9.203e+00</td>
</tr>
<tr>
<td>4.358e+00</td>
<td>1.126e+02</td>
<td>8.000e+00</td>
<td>3.947e+03</td>
<td>1.052e+01</td>
</tr>
<tr>
<td>4.680e+00</td>
<td>1.595e+02</td>
<td>1.250e+01</td>
<td>6.716e+03</td>
<td>1.182e+01</td>
</tr>
<tr>
<td>4.980e+00</td>
<td>2.123e+02</td>
<td>1.800e+01</td>
<td>1.024e+04</td>
<td>1.311e+01</td>
</tr>
<tr>
<td>5.290e+00</td>
<td>2.702e+02</td>
<td>2.450e+01</td>
<td>1.391e+04</td>
<td>1.430e+01</td>
</tr>
<tr>
<td>5.603e+00</td>
<td>3.335e+02</td>
<td>3.200e+01</td>
<td>1.772e+04</td>
<td>1.544e+01</td>
</tr>
<tr>
<td>5.922e+00</td>
<td>4.038e+02</td>
<td>4.050e+01</td>
<td>2.222e+04</td>
<td>1.661e+01</td>
</tr>
<tr>
<td>6.156e+00</td>
<td>4.809e+02</td>
<td>5.000e+01</td>
<td>2.750e+04</td>
<td>1.781e+01</td>
</tr>
<tr>
<td>6.393e+00</td>
<td>5.566e+02</td>
<td>6.050e+01</td>
<td>3.369e+04</td>
<td>1.904e+01</td>
</tr>
<tr>
<td>6.609e+00</td>
<td>6.574e+02</td>
<td>7.200e+01</td>
<td>4.094e+04</td>
<td>2.029e+01</td>
</tr>
<tr>
<td>6.810e+00</td>
<td>7.581e+02</td>
<td>8.450e+01</td>
<td>4.824e+04</td>
<td>2.154e+01</td>
</tr>
<tr>
<td>7.003e+00</td>
<td>8.613e+02</td>
<td>9.800e+01</td>
<td>5.627e+04</td>
<td>2.279e+01</td>
</tr>
<tr>
<td>7.196e+00</td>
<td>9.724e+02</td>
<td>1.125e+02</td>
<td>6.632e+04</td>
<td>2.401e+01</td>
</tr>
<tr>
<td>7.399e+00</td>
<td>1.088e+03</td>
<td>1.280e+02</td>
<td>7.420e+04</td>
<td>2.519e+01</td>
</tr>
</tbody>
</table>
SUBROUTINE INBUF (LI, Z, N, IEOF)

PURPOSE: TC TRANSMIT N WORDS OF DATA FROM DISK UNIT LI
        TO SCM ARRAY Z STARTING AT ADDRESS Z1C.

ARGUMENTS: LI ZINC - LOGICAL UNIT NUMBER OF DISK FILE
            Z IOUTC - FWA OF SCM ARRAY TO BE FILLED
            N ZINC - NUMBER OF WORDS IN Z TO BE FILLED
            IEOF IOUTC - END-OF-FILE FLAG

PROGRAMMER: G. ROOD, T-4

DATE: AUG 1976

******** SYSTEM DEPENDENT ********

DIMENSION Z(II)
INTEGER Z, FK, FK, FM
DATA FP, FP, FM/FM/
C----- IF FIRST TEST FOR END-OF-FILE
READ (LI, ENC=1) TEST
IF (TEST.EQ.FM) GO TO 1
C----- READ IN REQUESTED NUMBER OF WORDS
READ (LI, ENC=1) Z(I, I=1, N)
IEOF=1
RETURN
C----- END-OF-FILE WAS READ, SET FLAG
1 IEOF=0
RETURN
END

SUBROUTINE CUTBUF (LO, Z, N)

PURPOSE: TC TRANSMIT N WORDS OF DATA TO DISK UNL LO
        FROM SCM ARRAY Z STARTING AT ADDRESS Z1C.

ARGUMENTS: LC ZINC - LOGICAL UNIT NUMBER OF DISK FILE
            Z IINC - FWA OF SCM ARRAY TO BE TRANSMITTED
            N ZINC - NUMBER OF WORDS IN Z TO BE TRANSMITTED

PROGRAMMER: G. ROOD, T-4

DATE: AUG 1976

******** SYSTEM DEPENDENT ********

INTEGER Z, FK
DIMENSION Z(II)
DATA FP, FP, FM/FM/
C----- WRITE CLI REQUESTED NUMBER OF WORDS
WRITE (LC, (Z(I, I=1, N)
C----- WRITE CLI IN CASE NO MORE WRITES TO LO
WRITE (LC) FM
WRITE (LC) FM
C----- BACKSPACE OVER CLI TO BE READY FOR MORE WRITES
BACKSPACE LC
BACKSPACE LC
RETURN
C----- WRITE END-OF-FILE ONLY
1 WRITE (LC) FM
RETURN
END